FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Data Mining Tool for Sports Analytics

José Carlos Milheiro Soares Coutinho

DISSERTATION

Mestrado Integrado em Engenharia Informática e Computação

Supervisors: João Pedro Moreira

Second Supervisor: Cláudio Sá

July 25, 2019

Data Mining Tool for Sports Analytics

José Carlos Milheiro Soares Coutinho

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Doctor Armando Jorge Miranda de Sousa
External Examiner: Doctor Paulo Alexandre Ribeiro Cortez
Supervisor: Doctor João Pedro Carvalho Leal Mendes Moreira

July 25, 2019

Abstract

With the evolution of Real-Time Location Systems (RTLS), more affordable equipment is becoming available for tracking individuals with higher precision. This becomes an opportunity for researchers in the area of sports analytics, opening the possibility of extracting new knowledge related to player performance, movement patterns, among others. This knowledge, when applied to football or hockey players, provides the trainer with new insights which may be crucial for the improvement of the team's performance.

This dissertation project has the aim of creating a tool to allow sports trainers to easily understand how the team players are performing and to provide data scientists an easy way to employ data mining methods in match data. This tool uses positional data from football athletes, and can easily be extended to use data from other invasion-based team sports. After feeding the collected data to the system, feature extraction is applied to the data which can followed by running the off-the-shelf mining algorithms embedded in the system. The tool's results include several statistics and performance measures of players and are shown using appropriate data visualization techniques and easy-to-understand measurements so that a trainer can quickly understand the strengths and flaws of each player. Also, it includes an interface for running mining algorithms, such as association rules mining, subgroup discovery, and a new method for discovering frequent distributions.

In the end, the displayed results will allow trainers to acquire more knowledge about their players and make more informed decisions, possibly leading to improved player management and performance. Data scientists will also have an easy way to analyse sports data, which can translate into having insights about the data more quickly.

Resumo

Com a evolução de Real-Time Location Systems (RTLS), equipamento mais acessível fica disponível para localizar indivíduos com maior precisão. Isto torna-se uma oportunidade para investigadores na área de Sports Analytics, abrindo a possibilidade de extrair novo conhecimento relacionado com o desempenho de jogadores, padrões de movimentação, entre outros. Este conhecimento, quando aplicado a jogadores de futebol ou hoquéi, fornece o treinador com novas informações que podem ser cruciais ao melhoramento do desempenho da equipa.

Este projeto de dissertação tem o objetivo de criar uma ferramenta de data mining que permite a treinadores perceberem facilmente o desempenho dos jogadores da sua equipa bem como de fornecer a cientistas de dados uma maneira fácil de aplicar métodos de data mining em dados de jogos desportivos. Esta ferramenta utiliza dados posicionais de atletas de futebol, e pode ser facilmente estendida para outros desportos de equipa invasivos. Depois de inserir os dados recolhidos no sistema, é aplicada feature extraction nos dados, que pode ser seguida da execução de algoritmos de data mining embebidos no sistema. Os resultados da ferramenta incluem várias estatísticas avançadas e métricas de desempenho de jogadores, e são mostradas usando técnicas de visualização de dados apropriadas e medições fáceis de compreender, de modo a que um treinador possa rapidamente perceber os pontos fortes e fracos de um jogador. Para além disso, inclui uma interface para correr algoritmos de data mining, como association rules mining, subgroup discovery e um novo método para descobrir distribuições frequentes.

No final, os resultados mostrados irão permitir que treinadores obtenham mais conhecimento acerca dos seus jogadores e que façam decisões mais informadas, possivelmente levando a uma melhoria na gestão e desempenho dos jogadores. Cientistas de dados também terão uma maneira fácil de analisar dados de desporto, que se pode traduzir em obter informações acerca dos dados mais rapidamente.

Agradecimentos

Muito obrigado pai, mãe e avós por terem investido muito tempo e dinheiro para eu chegar até aqui. Muito obrigado aos meus orientadores, especialmente ao Cláudio, por aturar os meus cepticismos, por acreditar que eu consigo fazer numa hora o trabalho de uma tarde e por me ajudar a ser mais otimista. Muito obrigado ao Tio Quique por ter sido sempre a pessoa que me espicaçava com novos desafios. Muito obrigado à FEUP por ter sido a minha segunda casa. E muito obrigado aos meus amigos, por me terem sempre apoiado.

José Carlos Coutinho

Contents

1 Introduction **1**
- 1.1 Context and Motivation . . . 1
- 1.2 Objectives and Methodology . . . 1
- 1.3 Dissertation Structure . . . 2

2 Literature Review **3**
- 2.1 Clustering . . . 3
  - 2.1.1 EM and K-means . . . 4
  - 2.1.2 Clustering in spatiotemporal data . . . 5
  - 2.1.3 Minimum Entropy Data Partitioning . . . 5
  - 2.1.4 Subtrajectory Clustering . . . 6
- 2.2 Subgroup Discovery . . . 9
- 2.3 Data Analytics in Sports Data . . . 11
  - 2.3.1 Data Collection . . . 11
  - 2.3.2 Characteristics and Metrics . . . 11
  - 2.3.3 Data Mining Tools for Sports . . . 12
- 2.4 Association Rule Mining . . . 17
- 2.5 Summary . . . 17

3 Proposed Approach **19**
- 3.1 Frequent Distributions . . . 19
  - 3.1.1 Combining with Association Rules Mining . . . 21
- 3.2 UnFOOT . . . 22
  - 3.2.1 Data Processing . . . 23
  - 3.2.2 Embedded Data Analysis . . . 24
  - 3.2.3 User Interface . . . 25

4 Results and Experimental Setup **33**
- 4.1 Datasets . . . 33
  - 4.1.1 Source A Datasets . . . 33
  - 4.1.2 Source B Dataset . . . 34
  - 4.1.3 Electricity Dataset . . . 35
- 4.2 Frequent Distributions Analysis . . . 35
- 4.3 UnFOOT Analysis . . . 38
  - 4.3.1 Association Rules . . . 40
  - 4.3.2 Subgroup Discovery . . . 42

5 Conclusions and Future Work **43**
5.1 Conclusions 43
5.2 Future Work 43

References **45**

A Frequent Itemsets + Association Rules experiments **49**

B Dataset Examples **57**

List of Figures

2.1 Iteration of the minimum entropy data partitioning algorithm when trying to identify the formation of a football team. Each cluster represents one role in the team's strategic formation, and are shown in the figure as coloured circles (Image obtained from [BLC+14a]). 6
2.2 The Hausdorff distance between curves is small, while the Fréchet distance is large. 7
2.3 Curves f and g, and distance d on the left. On the right there is the resulting free space diagram. 8
2.4 GUI of the passing analysis tool, showing the passing possibilities and the passable areas, colored with the team color of the receiving player. Ball is represented in white 13
2.5 GUI for the passing sequence tool. This shows a passing sequence between player 1, 2 and 3, in which player 1 passes to player 2, which dribles during 4s until it passes to player 3. 14
2.6 Visualization of all representative subtrajectories of the clusters found for a central midfielder. It can be inferred that this central midfielder also executes corner kicks from the right side. 15
2.7 Correlation of the movement of three defenders. 16
2.8 Two teams' dominant regions (one in light gray, and another on dark gray). Each teams' players are represented by a different shape (triangle or square). 16

3.1 Diagram of the tools pipeline 22
3.2 Image of the Player view. 1 - Dropdowns to choose the players to compare; 2 - Radar chart with performance metrics; 3 - Simples statistics tables; 4 - Positioning information boxes; 5 - Charts with player metrics along time 27
3.3 Image of the Team view. 1 - Overall Score board; 2 - Graph display 28
3.4 Example of the Heatmaps display 28
3.5 Example of the Playing zones per player display 29
3.6 Image of the Data Analysis view 29
3.7 Example of the Frequent Distribution results of the tool 30
3.8 Example of the Association Rules results of the tool 31
3.9 Example of the Subgroup Discovery results of the tool 31
3.10 Image of the Settings view. 1 - File Upload board; 2 - Match Analysis board . . . 32

4.1 Speed profiles of players obtained in Experiment 5: Source A 36
4.2 Speed profiles of players obtained in Experiment 3: Source A 37
4.3 Profiles of the electricity prices. Note that distribution 2 scale is different from the others. 39
4.4 Speed profiles of the players in Match 2. 40

LIST OF FIGURES

A.1 Speed profiles of players in Experiment 1 49
A.2 Speed profiles of players in Experiment 2 50
A.3 Speed profiles of players in Experiment 3 51
A.4 Speed profiles of players in Experiment 4 51
A.5 Speed profiles of players in Experiment 5 51
A.6 Speed profiles of players in Experiment 9 53
A.7 Speed profiles of players in Experiment 10 53

List of Tables

3.1 Example Input File 22
3.2 Example of the new dataset after the initial pass over the data 23
3.3 Example of a player's positioning score. 24

4.1 Example of the electricity dataset records after the transformation 35
4.2 Most important rules found in Match 2: Source A 38
4.3 Association rules found in the Electricity dataset relative to the Victoria state: Experiment 9 38
4.4 Comparison between the real match results and the results of the tool 40
4.5 Number of players correctly classified by the position module in each match of Source A. 41
4.6 Best association rules of Match 1 and 2 41
4.7 Subgroups found in Match 1 and 2 with different targets 42

A.1 Association rules found in Experiment 1 49
A.2 Association rules found in Experiment 2 50
A.3 Association rules found in Experiment 2 51
A.4 Association rules found in Experiment 4 51
A.5 Association rules found in Experiment 5 52
A.6 Association rules found in Experiment 9 52
A.7 Association rules found in Experiment 10 for Team 1 54
A.8 Association rules found in Experiment 10 for Team 2 55

B.1 Example of the TXT file from Source B with the position measurements 60

LIST OF TABLES

Abbreviations

RTLS	Real-Time Location Systems
EM	Expectation-Maximization
RFID	Radio-Frequency Identification
MMT	Minimum Moving Time
DR	Distribution Rules
TSD	Time Series Dataset
EPTS	Electronic Performance and Tracking Systems
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Context and Motivation

In recent years, sports have been a focus on data analysis [GH17]. Computer vision technology has allowed recording data from the game without human intervention, and, in some leagues, devices for the recording of games are already mandatory. The availability of match data is not only beneficial for presenting statistics to the fans, but also to be used for further analysis. Sports analytics have used this data to gain better insights on players' and teams' performance [GH17], allowing teams to use that information to adapt training methods and game strategy for better results.

Developments in the area of Real-Time Location Systems (RTLS) are leading to equipment that is more accessible and reliable, which allows extracting higher precision data [NTO18]. Sports analytics researchers can use that data for extracting new knowledge related to team/individual performance, strategy effectiveness, *etc*. This knowledge, when combined with a proper dashboard, can provide trainers with new insights which may be crucial for the team's improvement.

1.2 Objectives and Methodology

The objective of this dissertation project was to develop a data mining tool for aiding in sports analytics. The tool provides football and hockey trainers with insights on how their team players are performing. The tool uses spatiotemporal data of matches and/or training. One part of the project was the application of feature engineering to the data. This is necessary before running the data mining algorithms implemented in the tool. A second part of the project was the implementation of a data mining module which included algorithms in the domain of subgroup discovery and spatiotemporal data mining. These have the objective of describing the players' and team's performance, as well as discovering other aspects of the team players that can be useful for the trainer. A third part of the project was the implementation of a data visualization module. This involved

experimentation on the best ways of transmitting information to sports trainers. The method of visualization should allow to quickly transmit the information obtained about the team, without needing anyone to interpret the data other than the trainer.

1.3 Dissertation Structure

Besides the introduction, this dissertation contains 4 more chapters. In chapter 2, the state of the art and related work is presented. Details about sports data and its features are described, as well as existing tools used in the analysis of sports data. Also, a description of data mining methods is included in this chapter. In chapter 3, it is presented a description of the tool that was developed in the dissertation project. All the necessary modules, including the one that uses a new method called Frequent Distributions, are detailed in this chapter. In chapter 4, the results are described and interpreted. At the beginning of this chapter is included a description of the datasets, as well as the transformations needed before using them. In chapter 5, we present the conclusions and possible future improvements.

Chapter 2

Literature Review

In machine learning, there are three types of learning: supervised, unsupervised and reinforcement. Supervised learning involves training a model providing inputs labeled with the corresponding classes. Then, the model learns how to associate the inputs to the corresponding classes, so that, when given a input with no class, it can predict to which class it belongs. Unsupervised learning involves models that try to find similarities between non-labeled inputs. Reinforcement learning involves finding the best sequence of actions or path that leads to the highest reward. This sequence is learned by trial and error and by having into account past results. The work in this project will focus on unsupervised learning methods, such as clustering and subgroup discovery.

2.1 Clustering

Clustering is an area of Data Mining that studies the division of data into groups of objects (clusters) that are *meaningful*, *useful* or both, based on the information found on the data [TSK05]. Objects in the same cluster should be related to one another, and should differ from the objects assigned to other clusters. Also, a measure to evaluate the difference between two objects must be chosen. (e.g. Euclidian distance, when the objects are points in space)

When trying to find meaningful clusters, objects with common characteristics are assigned to the same clusters, which helps on finding a classification for the objects in the data [TSK05]. Clustering has been used in different fields, such as medicine [SLT+18], business [DGH+18] and sports [BLC+14a]. When trying to find useful clusters, the objective is usually to find a cluster prototype, which is the representative of the cluster. Cluster prototypes can be used to reduce the number of individual objects to process in order to make the algorithms viable, or to compress the data [TSK05].
Tan, Steinbach and Kumar [TSK05] propose a classification for different types of clusterings:

- Hierarchical or Partitional: Partitional is the most simple approach, when objects are just divided into non-overlapping clusters. Hierarchical is when we have nested clusters, organised as a tree.

- Exclusive or Non-exclusive or Fuzzy: Exclusive is when an object can only be assigned to one cluster, as opposed to Non-exclusive, when it can belong to multiple clusters. Fuzzy clustering is when the belonging of an object to each cluster is measured between 0 and 1 (0 = certainly doesn't belong, 1 = belongs completely)

- Complete or Partial: Complete clustering implies that all objects are assigned to a cluster, as opposed to Partial, where objects may not be assigned to a cluster.

They also define several types of clusters:

- Well-separated clusters: Each object is more close to every other object in its cluster than to any object outside its cluster.

- Prototype-based clusters: Each object is more close to the centre/prototype of its cluster than to the centre/prototype of other clusters.

- Graph-based clusters: All objects in the cluster are interconnected, but have no connection to any object outside their cluster.

- Density-based clusters: Clusters are formed from a high density region of objects, and are surrounded by a region of low density of objects, which is considered as noise.

- Shared-property clusters: Clusters are formed by objects that share a common property. This is the generic definition of a cluster, which also includes the previous definitions.

The type of cluster should be decided according to the objective of the data analysis performed. For example, the second and third types tend to have a globular shape, which is not suitable to all situations, where should be considered the use of density-based clusters.

2.1.1 EM and K-means

One of the algorithms used in data mining approaches is the EM (*Expectation-Maximization*) algorithm. As Bishop explains in his book [Bis16], this algorithm is used for "finding maximum likelihood estimators in latent variable models". The EM algorithm fits in the clustering context since the objective of clustering can be to find an estimation of the classes' objects distribution. One of the instances of this algorithm is the K-means clustering [Bis16]. This technique is used to apply a complete exclusive clustering on the data which will partition it in K clusters, where K is assumed to be given in the beginning. Also, the technique uses a *distortion measure* which represents the distance of an object to the cluster's centre. The process begins with the creation of K points representing each cluster centre, generated randomly or by applying a initialization process. Then, the process follows 2 steps, as defined in the EM algorithm:

1. E-step: For each point, evaluate the distortion measure relatively to each cluster centre and assign the point to the cluster that resulted in the least distortion.

2. M-step: Calculate the new centres for each cluster, based on the new assignments obtained in the E-step.

Since random initialization can sometimes yield unsatisfactory results, Tan *et al.*[TSK05] give two examples of a better method for the initialization: Cluster centres can be initialized by performing several random runs, and choosing the one which produced better results; or by picking a small sample of the dataset, performing a run, and using the centres obtained as the initial centres for the whole-dataset run.

2.1.1.1 EP-MEANS

Henderson *et al.*[HGER15a] propose a method called EP-MEANS. This method clusters probability distributions regarding a target attribute. It is based in the K-means clustering algorithm ([Bis16]) and the Earth Mover's Distance ([RTG98]). In their approach, they start by picking a set of distributions of a target attribute. Then, they use the k-means algorithm to find the clusters of the set of distributions, using the Earth Mover Distance to measure the distance between distributions. This method can be non-parametric if we use automatic methods of finding the number of clusters, as described in the paper.

2.1.2 Clustering in spatiotemporal data

Spatiotemporal datasets contain spatial, temporal or spatiotemporal information. This information can be, for example, a registry of the location of earthquakes in the past years [MPS18], the GPS coordinates and timestamps of Tweets [LC18] or the recorded positions of vehicles [VBT09]. Using clustering methods in spatiotemporal datasets allowed, respectively, to predict the hotspots of earthquakes, to predict the home location of tweeter users or to discover flock patterns in vehicles. In this subsection, two clustering methods that are used in sports spatiotemporal data are described: Minimum Entropy Data Partitioning and Subtrajectory Clustering.

2.1.3 Minimum Entropy Data Partitioning

One of the approaches used to extract knowledge from spatiotemporal datasets is the Minimum Entropy Data Partitioning method [RER99]. We can view each cluster as a *probability density function (pdf)* which models the probability of an object belonging to that cluster, given the object's attributes [BLC+14b]. These probability densities may overlap, which means that the assignment of an object to a cluster may be similarly adequate for two different clusters. So, the objective of the method is to minimize the overlaps between all clusters. To measure the amount of overlap between two clusters, the Kullback-Lieber measure may be used [RER99, BLC+14b]. As shown by Roberts *et al.*[RER99], minimizing the total overlap is equivalent to minimizing the entropy of the clusters over all observed data. The less the entropy is, the more well-separated the clusters are. In [RER99], they propose that the minimization of the entropy should be made like a Radial-Basis function classifier. This means that:

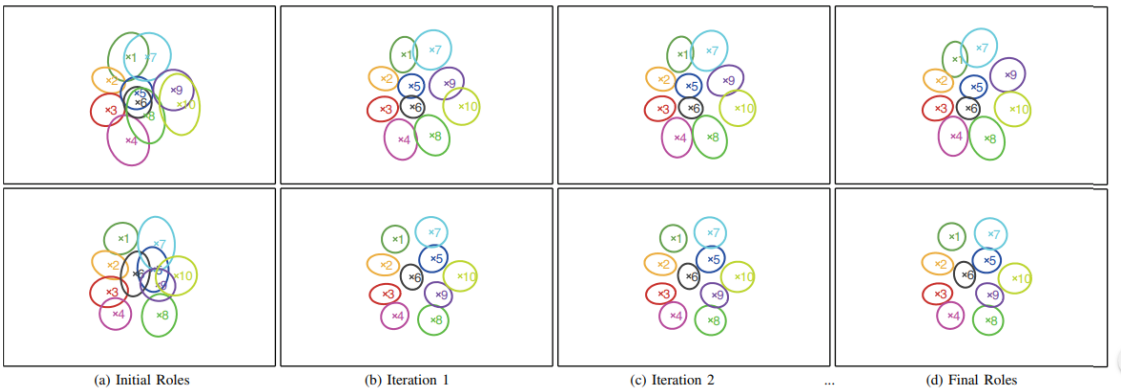


Figure 2.1: Iteration of the minimum entropy data partitioning algorithm when trying to identify the formation of a football team. Each cluster represents one role in the team's strategic formation, and are shown in the figure as coloured circles (Image obtained from [BLC+14a]).

- We start with a fixed set of pdf's.
- Each cluster is the weighted sum of the pdf's in the set.
- The classifier updates the weights in each iteration in order to minimize the entropy.

Bialkowski *et al.*[BLC+14b, BLC+14a] used a variation of the minimum entropy data partitioning method to identify football formations. The dataset used in their work comprises player tracking data for an entire season. They propose that a team has several unique roles (which will be our clusters), and that only one player can be in a role in each time-frame of the data. The method involves going through all time-frames and, for each of them, assign a player uniquely to a role, using the Hungarian Method [HGER15b]. Next, the method updates the roles' distributions in a way that minimizes the entropy between roles. The results reveal a 75.33% correct classification rate of the team's formations. A graphical example of the method can be found in Figure 2.1.

2.1.4 Subtrajectory Clustering

Another approach used to extract knowledge from spatiotemporal datasets is Subtrajectory clustering [BBG+11]. This method tries to find clusters of subtrajectories in a given set of trajectories. In order to cluster trajectories, we need a way to measure the distance between trajectories. Two different distance measures between trajectories are the *Hausdorff distance* and the *Fréchet distance*.

The Hausdorff distance between two curves P and Q ($\delta_H(P,Q)$) is expressed by the following equation [AKW04]:

$$\delta_H(P,Q) = max(\tilde{\delta}_H(P,Q), \tilde{\delta}_H(Q,P))$$

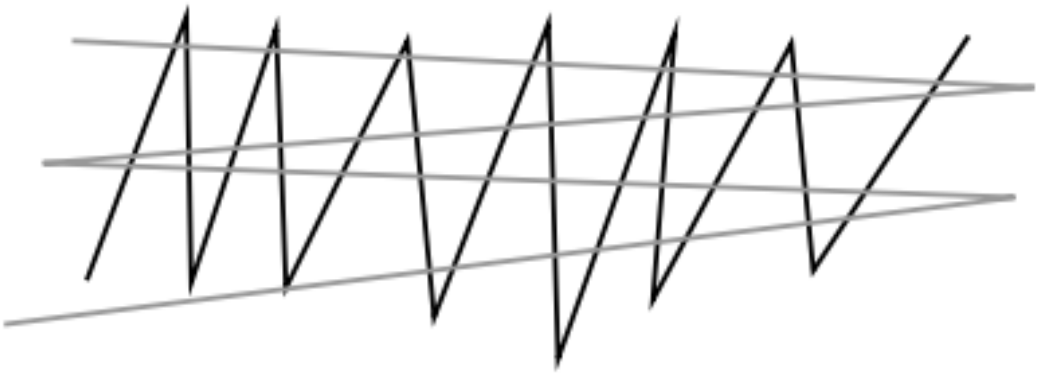


Figure 2.2: The Hausdorff distance between curves is small, while the Fréchet distance is large.

Where $\tilde{\delta}_H(P,Q)$, called *directed Hausdorff distance*, is defined as:

$$\tilde{\delta}_H(P,Q) = \max_{x \in P} \min_{y \in Q} \|x - y\|$$

In an informal way, the Hausdorff distance can be explained in two simple steps: First, for each point in the first curve, we obtain the minimum distance to the second curve. Then, amongst the list of minimum distances, we pick the maximum.

The Fréchet distance between two curves P and Q ($\delta_F(P,Q)$) is expressed by the following equation [AKW04]:

$$\delta_F(P,Q) = \inf_{\rho\,\sigma} \max_{t \in [0,1]} \|P(\rho(t)) - Q(\sigma(t))\|$$

ρ and σ range over continuous and non-decreasing functions with $\rho(0) = \sigma(0) = 0$ and $\rho(1) = \sigma(1) = 1$. Gudmundsson *et al.*[GW14] explains the concept intuitively: If we imagine a person walking his dog on a leash, the Fréchet distance is the smallest length of leash that allows the person and the dog to walk along their paths, while being able to change their speed or pause, but not backwards.
The advantage of the Hausdorff distance is that it is much easier to compute than the Fréchet distance. However, it does not have into consideration the course of the curves. This translates in having a distance that is too small for trajectories that do not resemble each other [AKW04] (See Figure 2.2). Since the Fréchet distance has better results considering trajectories, an approximated version for polygonal curves (called *discrete Fréchet distance*) is used in some works like [BBG+11] and [GW14]. On this version, instead of calculating the distance in all the points, it is only calculated in the vertices of the polygonal curves.

Another important definition is the *free space diagram* (See Figure 2.3) [BBG+11]. Consider we have the polygonal curves f and g, with n and m vertices, respectively. Also, let f be a polygonal curve with n vertices $p_1, ..., p_n$. We can define ϕ_f as a map: $[1,n] \longrightarrow \mathbb{R}^c$, where $i \in 1, ..., n$ maps to point p_i and c is the dimension of the points. The equation for the free space

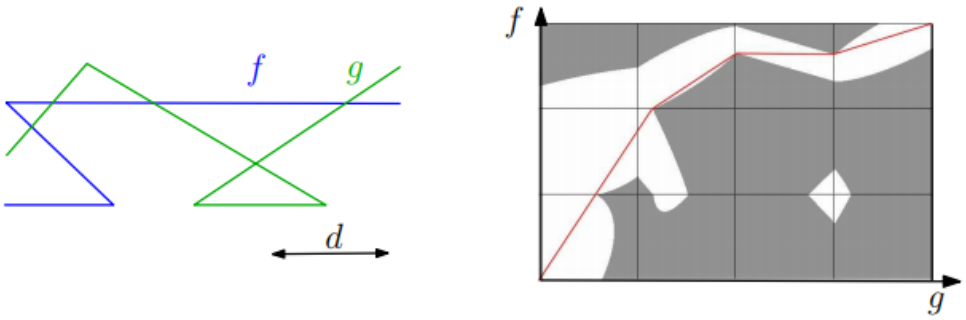


Figure 2.3: Curves f and g, and distance d on the left. On the right there is the resulting free space diagram.

diagram of curves f and g, with distance d ($F_d(f,g)$) is the following:

$$F_d(f,g) = \{(s,t) \in [1,n] \times [1,m] : |\phi_f(s), \phi_g(t)| \leq d\}$$

where (s,t) represents a tuple in the diagram. In this diagram, each axis represents each one of the curves, and going along the axis represents going along the corresponding curve. The white area represents the tuples in which the Euclidian distance between the points $phi_f(s)$ and $phi_g(t)$ is less or equal than d. Alt and Godau [AG95] showed that the Fréchet distance between f and g is less than d when there is a monotone path between the tuples $(0,0)$ and (n,m) in the free space diagram.

Now that we have the definitions in how to measure the distance between two trajectories, we can proceed to identifying the subtrajectory clusters. Buchin *et al.*[BBG+11] define three parameters in a subtrajectory cluster $SC(m,l,d)$. m is the number of non-identical subtrajectories in the cluster, l is the minimum length for any of the subtrajectories, and d is the maximum distance between subtrajectories. They also define two vertical lines in the free space diagram, l_s and l_t. In [BBG+11], they start by showing how to cluster subtrajectories in only one trajectory T. They say that there are m cluster curves between l_s and l_t in $F_d(T,T)$ such that:

- The m curves are not identical between each one of them, and all start at l_s and end at l_t
- Each curve is monotonically increasing in both coordinates from l_s to l_t
- The y-coordinates of two curves overlap in at most one point.

For the clustering algorithm, it is needed a data structure that stores a representation of the free space diagram as a directed labeled graph. Summarily, this algorithm consists of sweeping through F_d, and recording the clusters that are according to the parameters specified. To cluster for subtrajectories in a set of trajectories, only minor changes are needed [BBG+11]. Link all trajectories in one long trajectory, and only consider cluster curves which start and end in the same trajectory.

Huang *et al.*[HLW13] define a new trajectory pattern which they call frequent sub-trajectories with time constraints. In their experiment, they could discover patterns of this type in Tencent Microblog data sets by performing a variant of subtrajectory clustering that includes time constraints.

In the field of sports, Gudmundsson and Wolle [GW14] developed a tool that used Buchin *et al.*algorithm for analysing players' and ball movement in football. The objectives of this tool are reporting the most common patterns formed by the ball between defense and offense and report the most common movements of a player/group of players. In their experiments, they found out that it is harder to have clusters when the Fréchet distance is smaller. On the other hand, a large Fréchet distance increases the appearances of longer clusters. Also, Gudmundsson and Wolle [GW14] developed another tool to correlate the clusters found in the subtrajectory clustering tool. The general idea behind the tool is that players 1 and 2 have correlated movements if there are several subtrajectories in the clusters for 1 and 2 that overlap in time. They view each subtrajectory cluster as a set of time intervals, where the start and end times are the initial and end points of each trajectory in the cluster. They define a set of time intervals as *locally correlated* if there exists at least one point that is contained in every interval. Also, they define that k players are *globally correlated* if the number of local correlation sets λ between k clusters is greater or equal than a threshold θ. Formally, the definition of global correlation by Gudmundsson and Wolle [GW14] is:

- $T_1, ..., T_k$ are a set of k trajectory clusters
- θ is a positive threshold

$T_1, ..., T_k$ are globally correlated if there are $C_1, ..., C_\lambda$ local correlation sets which satisfy the following:

- $\lambda \geq \theta$
- $|C_m|_{1 \leq i \leq \lambda} = k$
- For each T_i, $1 \leq i \leq k$, and C_j, $1 \leq j \leq \lambda$, there exists a time interval ti such that $ti \in T_i \cap C_j$
- For all $1 \leq i \neq j \leq \lambda$, $C_i \neq C_j$

To discover this correlations, a sweeping-line approach is used.

2.2 Subgroup Discovery

Subgroup discovery is an area of Data Mining that studies ways of finding smaller exceptional models inside a general model. This exceptional models should be statistically "interesting", which means they should contain a number of samples that is sufficiently large, but at the same time being very distinct from the general model [Hel16]. These subgroups are represented by a group of attributes that produce unusual distributions when related. To measure how "interesting" a subgroup is, a *quality measure* is employed. Quality measures should have into account that a

subgroup is more interesting the more it is distant from the general group and the more samples the subgroup contains.

The general methodology for Subgroup discovery algorithms is described by Helal [Hel16]. Firstly, there is the phase of subgroup candidate generation. The objective of this phase is searching for new and more specific subgroups that may have a better quality measure value. Secondly, there is the pruning phase, where candidates are excluded based on a predefined criteria. Finally, there is the post-processing step, where the candidates are evaluated with a quality measure, allowing to distinguish which subgroups are the most interesting.

Helal [Hel16] also explains the three most common approaches for algorithms in subgroup discovery: *Exhaustive Search Based*, *Beam Search Based* and *Genetic Algorithm Based*. Exhaustive search based, as the name suggests, sweeps through all of the search space, verifying the suitability of each candidate. As expected, this is very resource intensive, and should only be used in small search spaces. Beam search based works iteratively, and is an alternative to Exhaustive search when the search space is too large. In this type of search, there is the concept of "beam", which is a limited-size container with the best partial solutions until that iteration. On each iteration, new candidates are generated from the candidates in the beam, by incrementing the number of constraints. Only the ones with the quality measure above a certain threshold are kept as candidates for the next iteration. Since this type of search does not explore all of the search space, it saves substantially more resources than exhaustive search, but has the disadvantage that a solution may not be found. Genetic Algorithm based uses the genetic algorithm heuristics, which are based on natural evolution. We start with a set of candidates with different descriptions and the corresponding quality measure value. From the starting set, the best candidates are chosen, based on their quality measures, and part of their descriptions are combined/switched, creating a new generation of candidates. Over the generations, there is an improvement of quality measure of the candidates, eventually getting to a solution.

Different pruning techniques can be used in subgroup discovery algorithms. Helal [Hel16] mentions three major types of pruning: Minimum support pruning, which involves removing the candidates that do not have enough samples in the dataset to support them; Optimistic estimate pruning; and Quality Constraint pruning, in which candidates which do not have more quality measure value than a certain threshold are removed. Also, different quality measures can be chosen for the algorithms, and the author mentions that unusualness and the Piatetsky-Shapiro measure are the most popular ones.

In [Hel16], Helal gives several examples where Beam Search based Subgroup Discovery was used in spatial data analysis, in diverse contexts. Also, in the context of sports, a Beam search based approach was used by de Leeuw [dLMK18] to define the pace profiles of runners that perform better in marathons and a subgroup discovery algorithm was used by Meerhoff [MdLGK] in football matches to try to discover the gameplay decisions of a team that lead to losses of ball possession.

2.3 Data Analytics in Sports Data

In recent years there is an increase in the collection of spatiotemporal data and the research in the sports area [BLC+14a, GW14, GH17, YLC+14]. In this section, a brief explanation will be given about which data is collected, and what are the features that are extracted from it. According to a recent survey (2017) [GH17], most of the match data regarding player trajectories and events logs is collected from football and basketball, which is reflected in the great quantity of research done in this two sports.

2.3.1 Data Collection

Nowadays there are a number of systems which can capture the spatiotemporal data from matches [GH17]. Leagues such as the NBA and the German Football League already capture data from all games, while other teams just capture the match data in their stadiums. Usually, these systems involve high-definition cameras positioned around the field which read the players' positions in real time, and in a post-processing stage, additional annotations such as faults are added manually or semi-automatically [GW14]. However, computer vision is not the only method of reading the players' positions. There exist also device tracking systems that make use devices attached to the players and/or ball/puck, which transmit the objects position using GPS or RFID technology.

2.3.2 Characteristics and Metrics

Spatiotemporal data extracted from matches has specific properties which are convenient in the field of research. In the case of team sports, players have an underlying structure, which corresponds to their formation/strategy, the number of agents is small since we only have the players, the ball, and possibly the referee, player positional data has a high sampling rate (at the time when [GW14] was published, the sampling rate was 10-25 samples per second) and have a small temporal and spatial range, and have a high interaction between agents [GH17]. Team sports data usually has two types of data collected: player/ball trajectories and event logs [GH17]. These two types of data can provide insights on the data individually, but they are best used in combination. For example, we can infer a team's formation from the player trajectories. However, this formation can differ whether the team is attacking or defending, which can be determined by the event log.

Player and ball trajectories are represented in the data as location points with a timestamp associated with it (relative to the match duration). From this data, we can directly infer, for example, the players' orientation and speed [GH17], or indirectly, for example, the maximum distance in high speed running [SRP17] or the player dominant region [TH00]. Event logs, as opposed to the player/ball trajectories, are not dense, but provide a lot of information, thus being very useful for inferring a number of aspects about the players and the team. Spatiotemporal data from entire competitions has some additional advantages besides the amount of matches and teams. With it, it is possible to run experiments on the weather conditions, on the fact that teams play at home/away or on injuries of players [GH17]. As Gudmundsson and Horton mention in their survey [GH17],

for team-based invasion sports such as rugby or American football, it is hard for computer vision systems to capture the spatiotemporal data from the matches. Since the number of collisions between players is very high in those sports, systems which rely on edge detection can have trouble identifying the players.

2.3.3 Data Mining Tools for Sports

There are several tools which make use of data mining approaches applied to sports. For example, there are already tools that analyse passing possibilities and sequences, player common trajectories and relationships between them and also the area that a player is guarding. In the examples presented, most of the tools were built with football data in mind, but they could be easily adapted to other sports.

2.3.3.1 Passing Analysis

One of the tools used by Gudmundsson *et al.*[GW14] has the objetive of evaluating football players' passing abilities. The passing abilites of a player involve the ability of executing a pass, the ability of receiving a pass, and the ability to detect when there is an opening for a pass. Also, in this tool, the definition of a pass includes the start coordinates, the initial speed, the direction of the pass and the player who performed the pass. They define the validity of the pass as when a player p makes a pass that can be reached by a player on the same team of p (besides himself) before anyone else. The tools receives as input 23 trajectories (1 for the ball and 22 for the players) and a passing speed, and outputs, for every point when a player has the control of the ball, the passing possibilities and the passable area for that given speed. This can be seen on Figure 2.4.

In this tool, is possible to choose the motion model of the players from 3 options: the first two are very similar to the ones proposed by Taki and Hasegawa [TH00] and Fujimura and Sugihara [FS05], and the third one is based on the historical data of a player. The motion models define how the player movement is described and what restrictions does it have. For example, in Taki and Hasegawa's approach [TH00], since the acceleration of a player is a constant from a set of possible accelerations, there is the possibility that the player's speed increases indefinately. In Fujimura and Sugihara's motion model [FS05], they include a resistive force that decreases the acceleration over time. The motions models are important to define the *dominant regions* of the players, which are the regions that one player can reach before any other player. This is explained in the section relative to the tool in which dominant regions were defined (Section 2.3.3.4). Gudmundsson *et al.*[GW14] mention that the results they got with the three different motion models were similar, so further research could be made to try to identify the differences,

The way the passable regions are defined in this tool is similar to the concept of dominant regions. Based on the player's position, speed and direction and the ball's position, the passable region of a player is the region where a player can receive the ball before any other player. Using the possible ball's positions in only discrete time steps, which gives a set of circles, they could

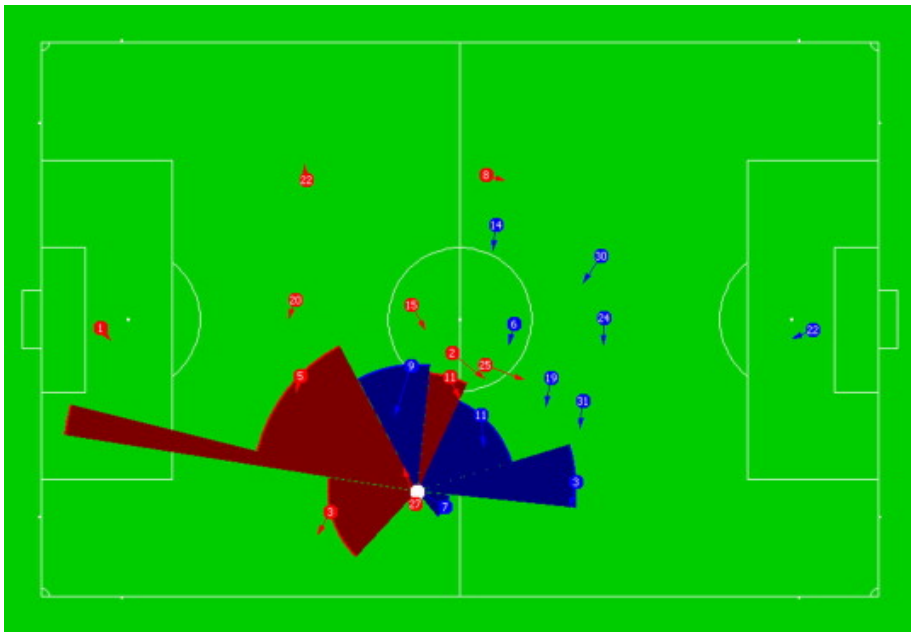


Figure 2.4: GUI of the passing analysis tool, showing the passing possibilities and the passable areas, colored with the team color of the receiving player. Ball is represented in white

intersect those circles with the players' passable regions to calculate the passable areas shown in the tool.

The authors mention that there is still the need to address more complex problems such as evaluating the player's passing ability, to recognize passing oportunities or to evaluate the player's receiving ability. Also, it is suggested that trying to address the problem from the defender's point of view would also be valuable for research.

2.3.3.2 Pass Sequence Analysis

Gudmundsson *et al.*[GW14] also developed another tool which analyses the pass sequences of one team. This is very useful to discover what are the most common patterns of passes when a team wants to make the a transition from the defense to the the attack or to discover players that have more interactions between each other. An example of the output is shown in Figure 2.5. The tools receives as input a set of players, a sequence of passes made by that set of players during several matches and a query (T, O), in which T is the number of players involved in that pattern, and O is the minimum number of passes that occured in the same sequence of T players. To perform these queries, a uncompressed suffix tree is built from the inputs. A character represents each player, and the team together forms a 11-letter alphabet. Also, a sequence of passes can be represented by a string of players which had possession of the ball sequentially, until the team lost the ball. Every edge of the tree corresponds to a character from that alphabet, and each node stores the frequency of the string that goes from the root until that node. After the tree is built, it is very easy to process a query. However, the building process is not immediate and, according to the

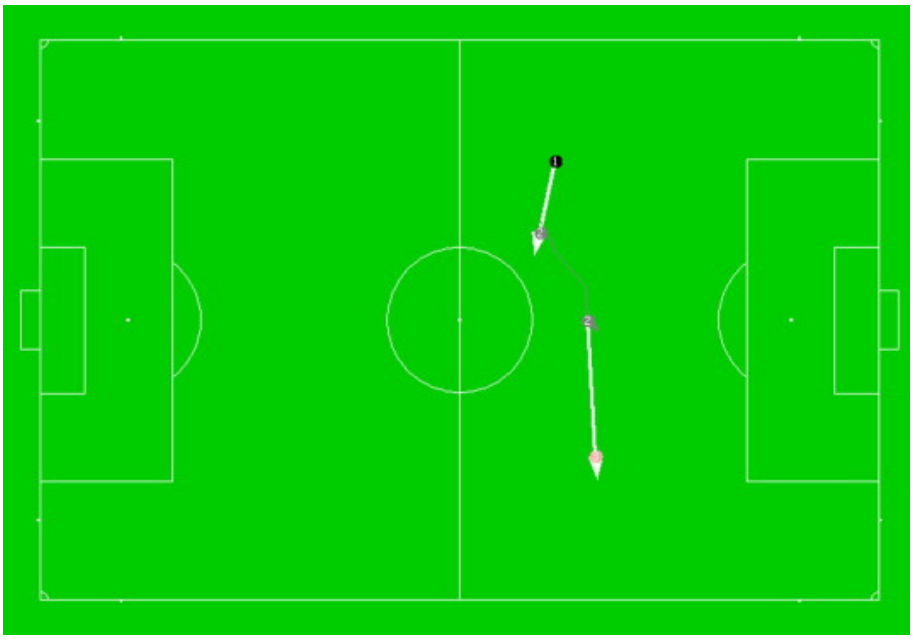


Figure 2.5: GUI for the passing sequence tool. This shows a passing sequence between player 1, 2 and 3, in which player 1 passes to player 2, which dribles during 4s until it passes to player 3.

authors, the process allocated around 2GB of internal memory for this task. For future work, they reccomend on optimising the space usage of the process.

2.3.3.3 Clustering and Correlating Subtrajectories

Again, Gudmundsson *et al.*[GW14] developed two other tools for analysing player movement. The first tool clusters player's movement and the second tool correlates the clusters obtained by the first tool.

The objective of the first tool is finding repeated movements of a player, by clustering subtrajectories of him according to the input parameters m (minimum number of subtrajectories), d (maximum distance between subtrajectories) and ℓ (minimum length of the subtrajectories). To group the subtrajectories in clusters, it is needed a distance measure between the subtrajectories. This will be further explained in Section 2.1. The measure used in this tool is the discrete Fréchet distance, which has into account the positional similarity between subtrajectories, but not their duration. This measure is also further explained in Subection 2.1.4. The algorithm used in the tool is the one defined by Buchin *et al.*[BBG$^+$11]. Figure 2.6 shows an example of the results of the tool. The authors mention that the parameters d and ℓ (especially ℓ) have a negative impact on the performance of the algorithm when increased. Since ℓ had the most impact, they filtered the vertices of the trajectories that were less crucial for the overall shape of the trajectory.

The second tool uses the subtrajectory clusters obtained by the first tool to find correlations between the most common movements of the players. First, the time interval when each subtrajectory happens is associated with its cluster. This means that each cluster of similar subtrajectories has a registry of the moments when that common movement happened. Then, the algorithm inserts the

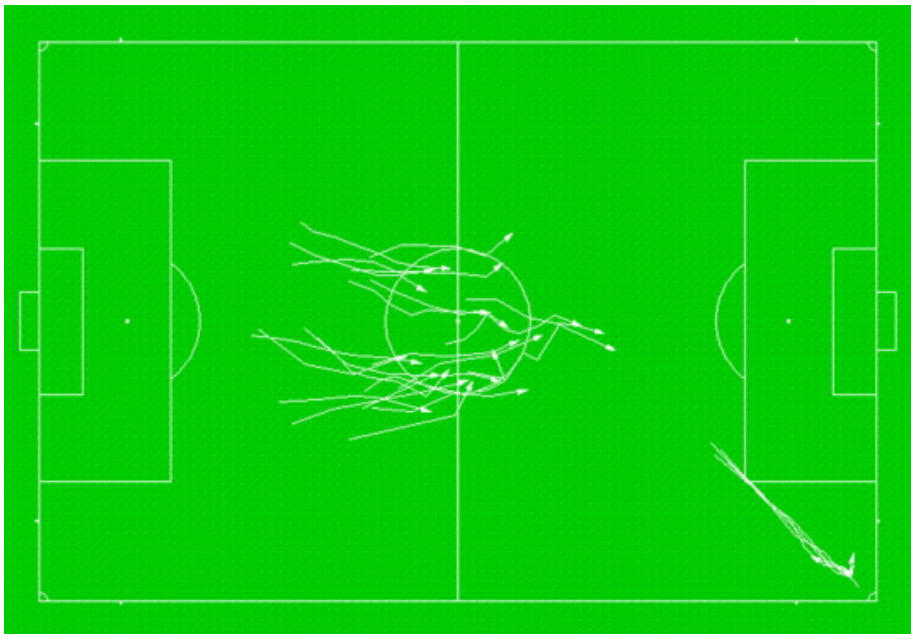


Figure 2.6: Visualization of all representative subtrajectories of the clusters found for a central midfielder. It can be inferred that this central midfielder also executes corner kicks from the right side.

subtrajectory clusters in a time line, sweeps through the timeline, and discovers a correlation between subtrajectories when the number of moments where both subtrajectories happened is higher than a certain threshold. This is explained in detail in Section 2.1.4. An example of the correlation between common movements of players is shown in Figure 2.7. Future work involves discussing the criteria for determining when the subtrajectory clusters are correlated.

2.3.3.4 Determining Dominant Regions

Taki and Hasegawa [TH00] presented a system which calculates the *dominant region* of players. They define the dominant region of a player as the region where a player can reach before any other player. To obtain the dominant region, we need to calculate the Minimum Moving Time Pattern (MMT). The authors define the MMT as the minimum time that is necessary for a player to go from his actual position to a certain point. To calculate the MMT we need the position, speed and an *acceleration ability* of the player. The latter is defined as "a set of acceleration patterns based on the physical ability of an average player" [TH00]. So, to calculate the dominant regions of players, we calculate the MMT of each player for each point and that point will belong to the dominant region of the player with the least MMT. Also, if we merge the dominant regions of the players of the same team, then we have the team's dominant region. The team's dominant region can be useful for evaluating the quality of a player's movement or pass. An example of two teams' dominant regions is presented in Figure X. This tool's calculation of the MMT assumes that players' acceleration patterns are the same for each player, and that their are constant, which is distant from reality. To address this second aspect, Fujimura and Sugihara [FS05] have proposed

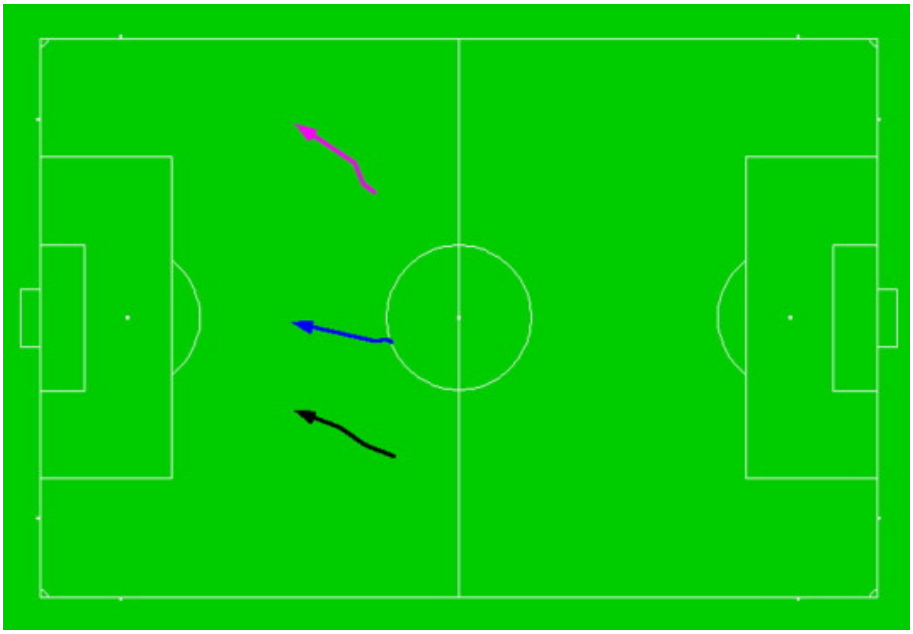


Figure 2.7: Correlation of the movement of three defenders.

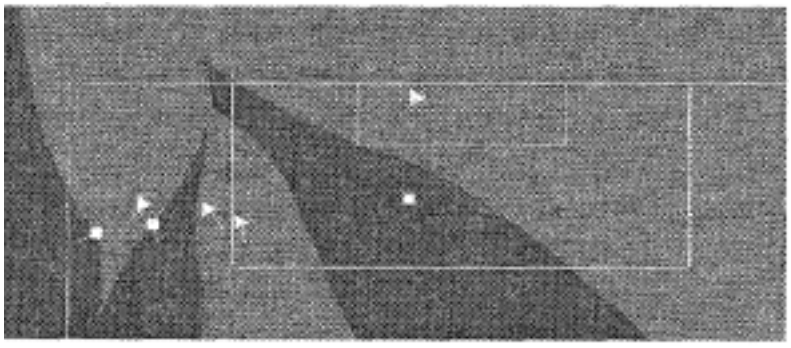


Figure 2.8: Two teams' dominant regions (one in light gray, and another on dark gray). Each teams' players are represented by a different shape (triangle or square).

a change in the calculation of the MMT which adds a resistive force that decreases velocity over time. This avoids that players can accelerate infinitely.

2.4 Association Rule Mining

Association rule mining is an area of Data Mining which studies ways of finding relationships between variables in a dataset. The relationships come in the form of implications $X \implies Y$, which we call *rules*. X and Y are itemsets, where X is usually called *antecedent* and Y called *consequent*. As an example, a rule found in a supermarket dataset could be $\{milk, butter\} \implies \{bread\}$. One way to measure how interesting a rule is, is to consider if the itemset $X \cup Y$ is observed frequently enough (which is called *support*) and if the rule is verified frequently enough (which is called *confidence*).

Jorge *et al.*[JAP06] propose a method called Distribution Rules (DR). Instead of discovering a relationship between two itemsets, DR discovers rules that associate a frequent itemset with a distribution of a target variable. The rules are represented in the form $A \implies y = D_{y\|A}$, where A is an itemset, y is the target variable and $D_{y\|A}$ is the distribution of y when A is observed. In this case, to measure how interesting a rule is, we check the difference between $D_{y\|A}$ and a reference distribution, which usually is the distribution of the whole population (D_y). In the paper ([JAP06]), they use the Kolmogorov-Smirnov statistical test [Con71]to measure the difference between distributions. This method is useful to avoid pre-discretizing a numeric target variable, which, in turn, avoids loss of information.

2.5 Summary

To conclude, there has been significant research in data mining applied to sports, because of the rise in interest for sports analytics. Most data mining tools in the area of sports focus on football and have promising future uses if further research is made to calculate more metrics that were not included in the tools due to their complexity. Also, there is evidence that clustering, association rule mining and subgroup discovery in spatiotemporal data can be used to extract new knowledge about the performance of teams and their players.

Chapter 3

Proposed Approach

The proposed solution is called UnFOOT: a data mining tool that receives players' spatiotemporal data from a match and displays an interactive interface for analyzing the data.[1] The tool shows a summary of each player's performance and the overall statistics of each team. It also allows for further analysis of the match using a data mining module. This module includes subgroup discovery and association rule mining methods as well as a new method called Frequent Distribution mining. The tool is intended to be used by trainers, sports analysts and data scientists. It includes plots and graphs that are easily interpreted by anyone with knowledge in sports, but also include interfaces that data scientists can use to run more advanced data mining methods.

3.1 Frequent Distributions

Time series data is constituted by a set of records, each of them recorded at a specific time [BD13]. Each record can contain values of different variables. Also, multiple entities can be represented. This means that each entity will have multiple records associated, and the variables in the records will represent the entity's attributes. Let us define a time series dataset (*TSD*) as a table with n rows and $m+2$ columns. The columns come in the format $cols = \{attr_1, attr_2, ...attr_m, t, ent\}$. We call $A = \{attr_1, attr_2, ...attr_m\}$ the *attributes*, $E = \{ent_1, ..., ent_k\}$ the *entities*, where k is the number of entities in the *TSD*, and $T = [t_{first}, t_{last}]$ the *timestamps*, where t_{first} and t_{last} are the first and last timestamps registered in the *TSD*. Each row r represents the record of the attributes for one of the entities for a timestamp t. More formally, $r_{t,enti} = A_{t,enti} = \{attr_1, attr_2, ...attr_m, t, enti\}, i \in [1,k]$.

Looking for ways of extracting information about the behavior of these attributes can be very important for the analysis of the data. One of the aspects that we may be looking for is the existence of patterns on the variability of the attributes. We can call this variability patterns as *profiles*. A profile (pf) is represented as a distribution of the values of a given attribute during a time interval ($dist_{attr}(t_{start}, t_{end}), t_{start} < t_{end} \wedge t_{start}, t_{end} \in T$). A *TSD*will have a fixed set of profiles for a given attribute ($PF_{attr} = \{pfattr_1, ..., pfattr_n\}$) and the variability of an attribute can switch

[1] A video with a demonstration of the tool can be watched in https://www.youtube.com/watch?v=x86tg48qEs4

between the different profiles over time. For example, when looking at records of electricity consumption, we will probably have $PF_{consumption} = \{pfconsumption_{day}, pfconsumption_{night}\}$. For $pfconsumption_{day}$, the distribution will include lower values, as opposed to $pfconsumption_{night}$, which will include higher values due to the need for artificial lighting during the night. Finding these profiles can be important for evaluating the different types of behavior an attribute can have. The process can be compared to clustering since we are trying to aggregate data by finding a representative (profile) for each group (a type of behavior) [HGER15a].

As mentioned in Section 2.4, Jorge *et al.*[JAP06] proposed an approach for mining distribution rules of a target given a set of attributes. In their approach, they choose a target attribute, which we will call *target*. Then, they apply association rule mining to obtain rules which match a set of attributes to the distribution of target. This distribution should be different enough from the whole distribution of target so that we do not have rules which do not add any additional information about the data. Using this method, we can obtain a distribution of target that stands out of the whole distribution, given a set of attributes. However, this is not useful to establish profiles. In order to find a profile, similar distributions of the target must be observed repeatedly throughout the time series. Also, we should be able to match each observation of target to one of the profiles. With the distribution rules method, we can obtain a distribution of target that stands out from the whole distribution, but there is no way of determining if that distribution is observed repeatedly in different time intervals.

In Subsection 2.1.1.1, is mentioned that Henderson *et al.*[HGER15a] have another approach which clusters distributions. Their approach uses the K-Means clustering method ([Bis16]) and the Earth Mover's distance ([RTG98]) to obtain the clusters of distributions of the target. In the end, each cluster will include the distributions of the target that belong to one profile, and the centroid of each cluster will be the representation of the corresponding profile. However, the process of clustering involves iterating through the data multiple times, which could be time consuming.

The Frequent Distribution mining method is an approach which iteratively discovers new profiles in the time series data. First, we choose a target attribute (*target*), the size of the time windows (*wsize*) and a distance threshold (*threshold*). The last 2 parameters will be explained further on. The values for the target can be recorded for one or multiple entities at the same time. Formally, $dist_{target}(t_{start}, t_{end}) = \sum_{i \in E} \left(dist_{target}(t_{start}, t_{end}, ent_i)\right)$. We divide the time series into time intervals with *wsize*, called *time windows* and we go through them sequentially. In each time window (t_{start}, t_{end}), we observe $dist_{target}(t_{start}, t_{end}, ent)$ for each $ent \in E$ and try to assign each $dist_{target}$ to a profile. This is done by checking the *distance* between the distributions and the discovered profiles using the Kolmogorov-Smirnov statistical test. The distribution is assigned to a profile if $distance < threshold$. Both *distance* and *threshold* are in the range $[0,1]$. If $distance(dist_{target}, pftarget_a) \geq threshold, \forall pftarget_a \in PF_{target}$, then it means that a new profile was found. When that happens, $PF_{target} = PF_{target} \cup \{dist_{target}\}$. The pseudocode for Frequent Distributions is shown in Algorithm 1. With this method, by making small additions, we can record the number of distributions that were assigned to each profile, and which profiles were assigned to the distributions of one entity, and even on which time windows a profile was assigned to a

distribution of an entity.

Input: target, wsize, threshold
begin
  profiles_list;
  foreach *time_window* = $(t_{start}, t_{end}), t_{end} - t_{start} = wsize \wedge \{t_{start}, t_{end}\} \in T$ **do**
    foreach *ent* **do**
      $entity\_distribution \longleftarrow dist_{target}(t_{start}, t_{end}, ent)$;
      $is\_distribution\_distinct \longleftarrow True$;
      foreach *profile in profiles_list* **do**
        if $distance(entity\_distribution, profile) < threshold$ **then**
          $is\_distribution\_distinct \longleftarrow False$;
        end
      end
      if *is_distribution_distinct* **then**
        Add *entity_distribution* to *profiles_list*;
      end
    end
  end
  return *profiles_list*
end

Algorithm 1: Frequent Distributions algorithm

3.1.1 Combining with Association Rules Mining

We can combine the Frequent Distributions with Association Rules mining by adding extra steps to the method. The objective is to obtain association rules that, for each entity, measure the transition between profiles in consecutive time windows. For example, it would be interesting to observe that, for a given target attribute, a profile A is always followed by a profile B in the next time window.

In order to do this, first, we have to have a record, for each entity, of the profile that was observed in each time window. Afterward, we need to find the frequent itemsets that will be used in the association rules mining. Each itemset will be composed of a pair of consecutive profiles: a previous profile and a next profile. So, we need to scan through the time series and obtain, for each consecutive time windows, the pair of consecutive profiles. After obtaining all the itemsets in the last step, we can run off-the-shelf algorithms to find the itemsets that are frequent and afterward use association rules mining. An itemset is considered frequent if the support for that itemset is higher than the user-defined minimum support. Also, in our approach, an association rule will only be considered relevant if its confidence value is higher than the user-defined minimum confidence threshold.

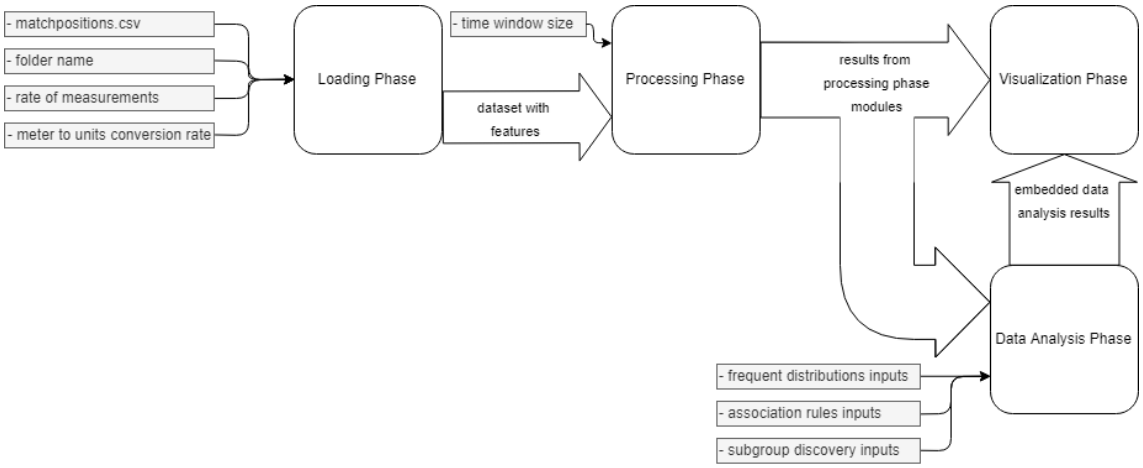


Figure 3.1: Diagram of the tools pipeline

3.2 UnFOOT

As previously stated, the UnFOOT tool receives spatiotemporal data from a match and displays an interface which allows the analysis of the data. The data has to come in a specific format so that the tool is able to use it. A list of requirements for the data is presented below:

1. The data should come in a .csv file with the format presented in Table 3.1. The *Player_id* should be an integer value. From 0 to 23 are players from the 1st team, and from 24 to 39 are players from the 2nd team. *Period* should be 1 or 2 and refers to the half of the match when that record was obtained (1 for 1st half, 2 for the 2nd half). *Timestamp* is the time elapsed since the beginning of the period. This does not have to be in seconds. The conversion rate from timestamps to seconds has to be specified when loading the file, in the appropriate input box. *x* and *y* are the coordinates of the player in the field, where x represents the width and y the length. They do not need to be in meters. The conversion rate from meters to the units of the xy data has to be specified when loading the file too.

Table 3.1: Example Input File

ID	Player_id	Period	Timestamp	x	y
1	2	1	15	-37	-45
2	2	1	16	-35	-49

2. The rate of the measurements should be constant. If the rate is 1 Hz, then there should always be a measurement each second, even if the measurement includes only null or NaN values.

The tool requires to have Python 3 installed, as well as the necessary modules.

3.2.1 Data Processing

The pipeline of the tool starts with the loading and preprocessing of the file, which we call the loading phase. In the loading phase, the user inputs the name for the match, the rate of the measurements in Hz, the conversion rate from meters to units of the xy data and the file itself. After loading the file, the tool makes one pass on the data. During this pass, the referees and the ball records are filtered, if they exist. Also, the tool adds the team number to the player records, calculates and adds the velocity, acceleration, and distance covered to the records, and fills the null/NaN values on the xy data by applying a linear interpolation between the records that are not null/NaN. At the end of the loading phase, the new dataset with the extracted features has the format shown in Table 3.2. In the next step, called the processing phase, the user has to choose a time window

Table 3.2: Example of the new dataset after the initial pass over the data

ID	Player_id	Period	Timestamp	x	y	team	Dist	Dist_x	Dist_y	v	v_x	v_y	a	a_x	a_y
28	1	1	10	-48.9	-0.3	1	0.1	0.1	0.0	1.0	1.0	0.0	9.99	-9.99	0.0

size, in seconds. The new dataset is divided into windows of that size. For each window, several internal modules extract different performance indicators and statistics from the positional data. The results are stored in each module while going through all the time windows. The modules are described below:

- Speed Module: Measures the player's ability to reach higher speeds. In each time window, obtains the maximum speed of each player and divides the speeds into quartiles. Each quartile gives a score. The higher the speed, the higher the score. In each time window, the obtained score for each player is added to the global score registered until then.

- Stamina Module: Measures the player's ability to run for longer distances. In each time window, obtains the distance that each player traveled in that time interval and divides the distances into quartiles. The higher the distance, the higher the score, similar to the Speed module.

- Agility Module: Measures the player's ability to change his trajectory. In each time window, for each player, calculates the agility score for each pair of consecutive positions and sums the scores. The agility score is calculated with the product of two values: the angle between the speed vector of the initial and final positions; the absolute value of the acceleration. The higher the player has changed direction and the higher the acceleration while changing, the higher the score will be.

- Pressing Module: Measures the number of times a player spent near players from the opposite team. The assumption behind this module is that whenever players of different teams are close together, they are pressing each other. For each instant, we cluster the players according to their position using DBSCAN. The score for each player increases whenever the player is included in a cluster with players from both teams. All the clusters that contain

players from only one team are excluded since players from the same team do not press each other. In the end, the more frequently the player was near the other team, the higher his score will be.

- Positioning Module: Measures the positioning of the player relative to his team. In each time window, for each team, equally divides the players' positions into 3 sections, both along the length and width of the field. Along the length, we have the sections Defense, Midfield and Attack. Along the width we have Left, Center and Right. The number of occurrences in each section will be counted for each player. In the end, the score of a player will look like what is shown in Table 3.3. In this case, we can infer that this player is a defender which plays more on the left side. This score is always relative to the team's overall position. Even

Table 3.3: Example of a player's positioning score.

Def	Mid	Atk	Left	Center	Right
8	2	0	6	4	0

if a team is playing on the attack, the players that are playing more on the back will be counted as being on the Defense section, even if their absolute position is, for example, the middle of the field.

- Simple Statistics Module: Calculates simple statistics and measurements for each player in the whole match. Includes the player's maximum speed, average speed, total distance traveled, maximum acceleration, average acceleration, total time played, and total time running. We consider that a player is running when his speed is above 12 km/h.

- Heatmap Calculator: Transforms the field in a $w \times l$ rectangle grid, where w is the number of rectangles along the width, and l the number of rectangles along the length. The Heatmap Calculator registers the players' positions that fall in each rectangle of the grid. This result will be displayed in the Team view, which is going to be explained in Section 3.2.3.

- Overall Score Calculator: Calculates the total score of the players. This module gets the scores from the Speed, Stamina, Agility and Pressure modules and normalizes them. Then, it calculates the overall score for each player by calculating the mean of the 4 scores and multiplying by 100.

3.2.2 Embedded Data Analysis

Besides the modules that were mentioned before, there are additional modules which are only used in further phases of the pipeline. These modules contribute with an additional layer of analysis by introducing data mining algorithms in the tool. The modules are described below:

- Association Rules Mining Module: Uses association rules to obtain relationships between consecutive time windows for each player. Before running the algorithm, the module loads

the results dataset. This results dataset is obtained after the processing phase, and each row r represents the scores for each player p in one time window t. The module gets from the results dataset a list of all the pairs $(r_{t,p}, r_{t+1,p})$, and labels all the $r_{t,p}$ as *prev* and the $r_{t+1,p}$ as *after*. It will also pick the values for Agility and group them in quartiles, for simplicity. Using the Apriori algorithm [AS94], the module will interpret the pairs in that list as itemsets and find the ones that are frequent. An itemset is considered frequent if $support(itemset) > min_{support}$, where $min_{support}$ is defined by the user. Then, it calculates the association rules using those frequent itemsets. We used the package mlxtend[2] to make the calculations. A rule is only accepted if $confidence(rule) > min_{confidence}$, where $min_{confidence}$ is defined by the user, and if the items in the antecedents and consequents are labeled as *prev* and *after*, respectively. In the end, we expect this module to find patterns which show, for example, that if a player reveals a high-speed performance in a time window, he will probably reveal a low-speed performance in the next one.

Instead of using this method, we could have used the Sequential Pattern mining method proposed by Ayres *et al.*[AFGY02]. However, we were only interested in the immediate transitions, so there was no need to build a full lexicographic transition tree, as required by the method of Ayres *et al.*. We opted by using our method because it was simpler and sufficient for this case.

- Subgroup Discovery Module: Obtains subgroups with unusual behavior relatively to a user-defined target. The module receives a target attribute (*target*), a target value (*target_value*) and an indication if higher or lower. Then, it will binarize the target attribute in the dataset. This is done by setting $r_{t,p}[target] = True$ if $r_{t,p}[target] \geq target\_value$ or $r_{t,p}[target] \leq target\_value$, depending if the indicator is > or < respectively. After the binarization, the module will perform the subgroup discovery task using the Chi-Squared quality function [Hel16]. The search for the subgroups is done using beam search (explained in Section 2.2). The module will store the subgroups' descriptors as well as the quality function score. For the subgroup discovery task, we used the python module pysubgroup[3]. In the end, we expect to obtain, for example, that usually, when a player reveals that his speed score is high, the player is playing in the attack and on the left side of the field.

- Frequent Distributions Module: Obtains the profiles of a specific attribute. This module is already described in Section 3.1. In the end, we expect to obtain, for example, that three profiles of player speed were found, one with low speeds, one with medium speeds, and one with high speeds.

3.2.3 User Interface

Visually, the tool is composed of 4 different views: Player, Team, Data Analysis, and Settings. The functionalities in these views are explained below:

[2] http://rasbt.github.io/mlxtend/
[3] https://pypi.org/project/pysubgroup/

- Player view: Allows to visualize details about each player's performance as well as comparing players. The content displayed in the view was obtained in the processing phase. In the top of the Player tab, there are 2 dropdowns (number 1 in Figure 3.2), one on the left and one on the right, where the user can choose the players to analyze and compare. We will call the player chosen on the left dropdown by *leftplayer* and the player chosen on the right dropdown by *rightplayer*. The dropdowns also show the overall score of each player, obtained by the Overall Score Calculator (Subsection 3.2.1). Besides the dropdowns, the view has 4 main components. The first one is the radar chart (number 2 in Figure 3.2). Each corner of the chart represents one of four player *performance metrics*: Speed, Stamina, Agility and Pressure. The method for obtaining these metrics is described on subsection 3.2.1, in the respective modules. The blue area corresponds to the left player's performance metrics and the orange area to the right player's performance metrics. The second one is the statistics table (number 3 in Figure 3.2). This table shows, for the players that the user chose, the measurements and statistics extracted with the Simple Statistics Module (Subsection 3.2.1). The table on the left and the table on the right show the measurements and statistics for the left player and right player, respectively. The third one is positioning information box (number 4 in Figure 3.2). It shows the percentage of the player's positions that belong to each label (Attack, Midfield, Defense, Left, Center, Right), the overall score of the player, which is the same shown in the dropdowns, and the estimate of the player's role in the team. The role estimate is made by comparing the values of each label or checking if the values are above certain thresholds. The input data for this component is the Positioning module results (Subsection 3.2.1). The fourth one is the plot of the player metrics along the time of the match (number 5 in Figure 3.2). In the upper section of this component, there is a dropdown where the user can select between one of the four player performance metrics. After the selection, a line plot with the values for the selected metric is shown.

- Team view: Allows to visualize aspects about the performance of each team and compare them. The content displayed in this view was also obtained in the processing phase. In the team tab, there are 2 main components. The first is the overall scoreboard (number 1 in Figure 3.3). In this board is shown the overall score of the two teams and the three best players of each team and their overall scores. The overall score of each team is the sum of the overall scores of the players of that team. The second component is the graph display (number 2 in Figure 3.3). There is a dropdown on the upper section of this component to choose one of several options: Team Positions, Speed, Stamina, Agility, Pressure, Playing zones per player. When the user chooses the Team Positions option, a heatmap representation of each team's positions is shown (Figure 3.4). Red tiles represent higher values and blue tiles represent lower values. The size and input values for the heatmap are obtained from the results of the Heatmap Calculator module (Subsection 3.2.1). When the user chooses one of the options Speed, Stamina, Agility or Pressure, a histogram of the chosen performance metric will be shown (number 2 in Figure 3.3). The histograms have bins of size = 10, with the

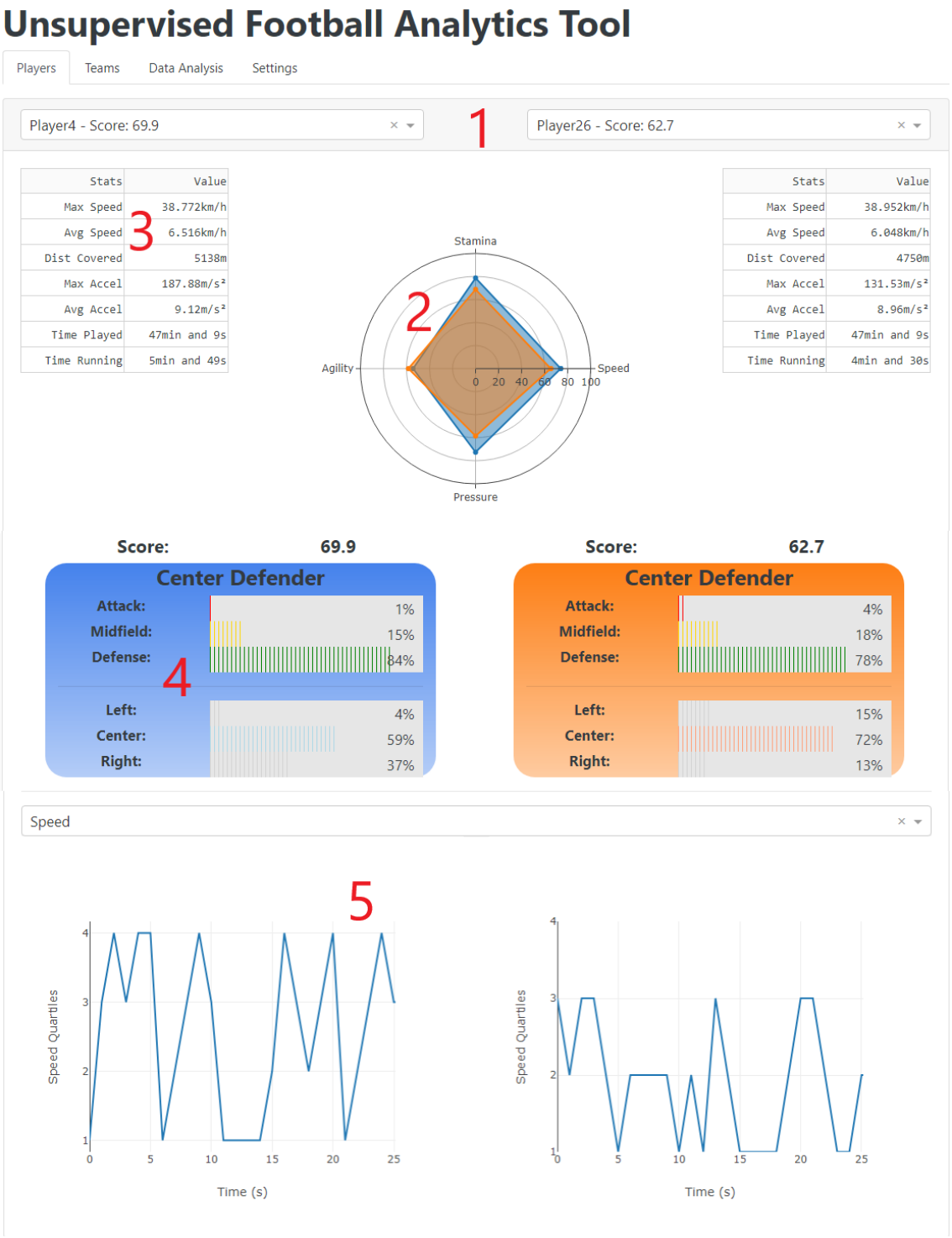


Figure 3.2: Image of the Player view. 1 - Dropdowns to choose the players to compare; 2 - Radar chart with performance metrics; 3 - Simples statistics tables; 4 - Positioning information boxes; 5 - Charts with player metrics along time

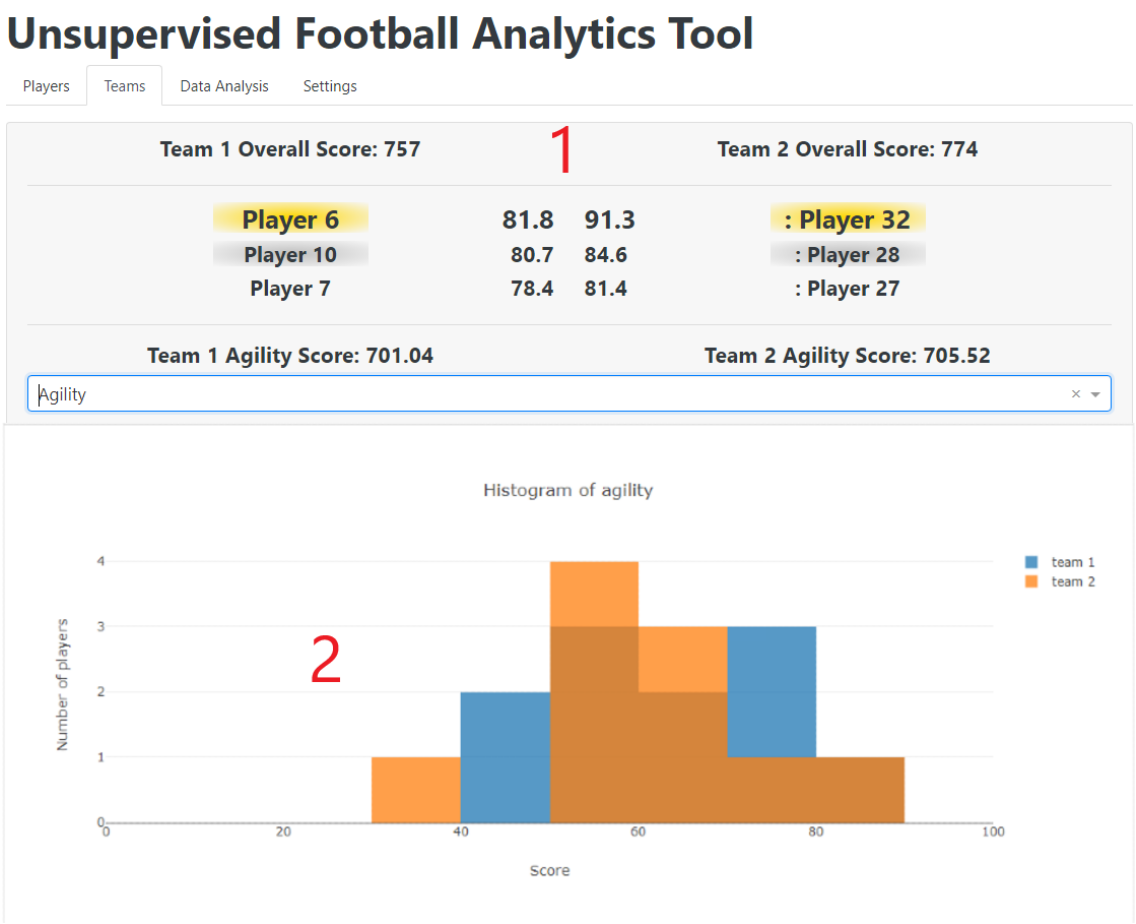


Figure 3.3: Image of the Team view. 1 - Overall Score board; 2 - Graph display

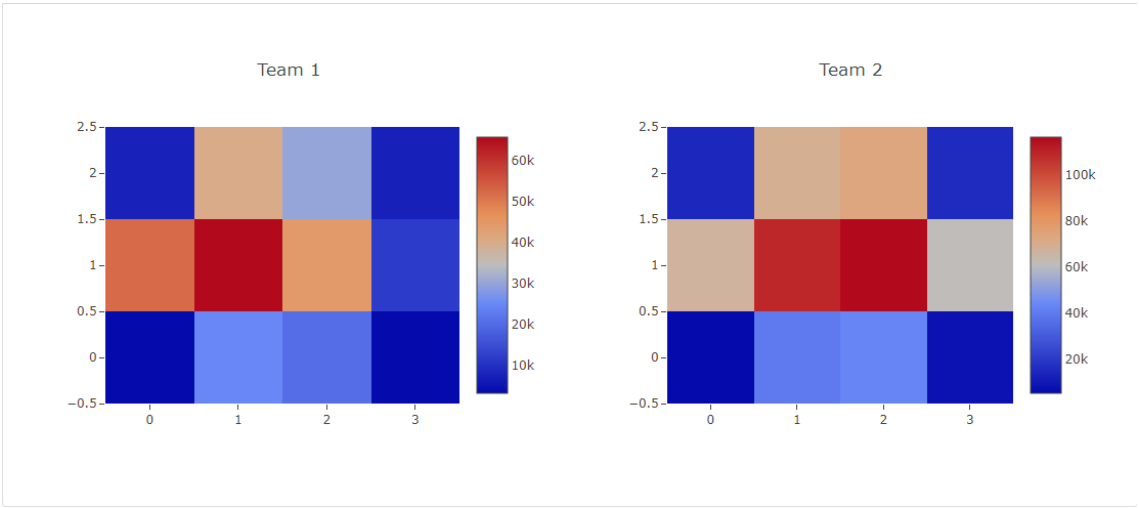


Figure 3.4: Example of the Heatmaps display

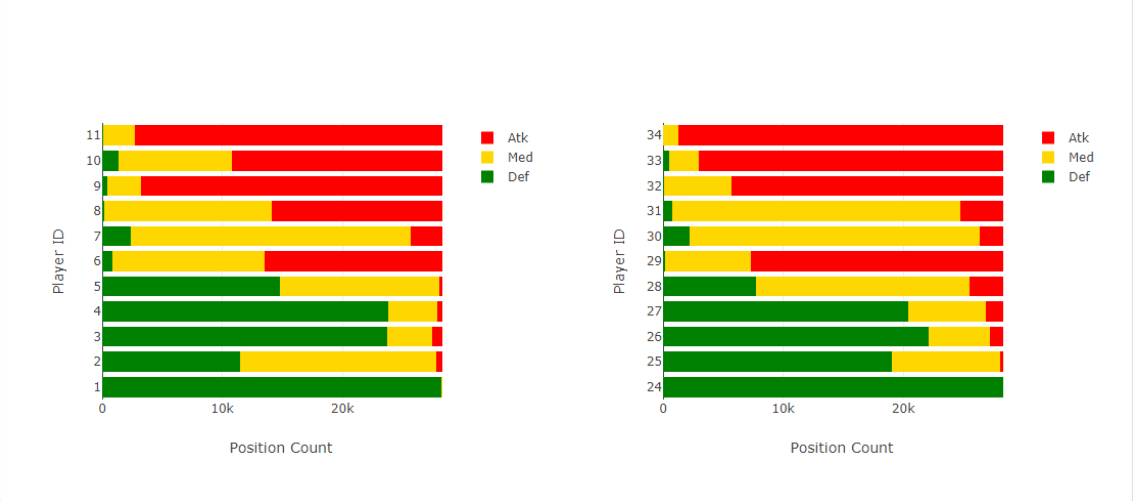


Figure 3.5: Example of the Playing zones per player display

blue bars representing team 1, and the orange bars representing team 2. Each bar represents the number of players of that team whose score falls on that bin. Also, the teams' scores for the chosen performance metric will be shown above the dropdown. When the user chooses the Playing zones per player option, a horizontal bar chart is displayed per team. Each bar corresponds to a player. The values for the bars are obtained from the results of the Positioning module (Subsection 3.2.1). The green part corresponds to the Defense score, the yellow part to the Midfield score and the red part to the Attack score.

- Data Analysis view: Allows to run the modules described in Subsubsection 3.2.2. It has 3 separators, as shown in Figure 3.6. Each separator as a set of inputs, a button to start running the analysis and a button to show the results. The first separator runs the Frequent Distributions module. From left to right, there is a dropdown to select the target attribute, an input box to select the size of the time windows, another input box to filter the results by a minimum support and a slider bar to select the distance threshold. The second separator runs the Association Rules Mining module. From left to right, there is a dropdown to select

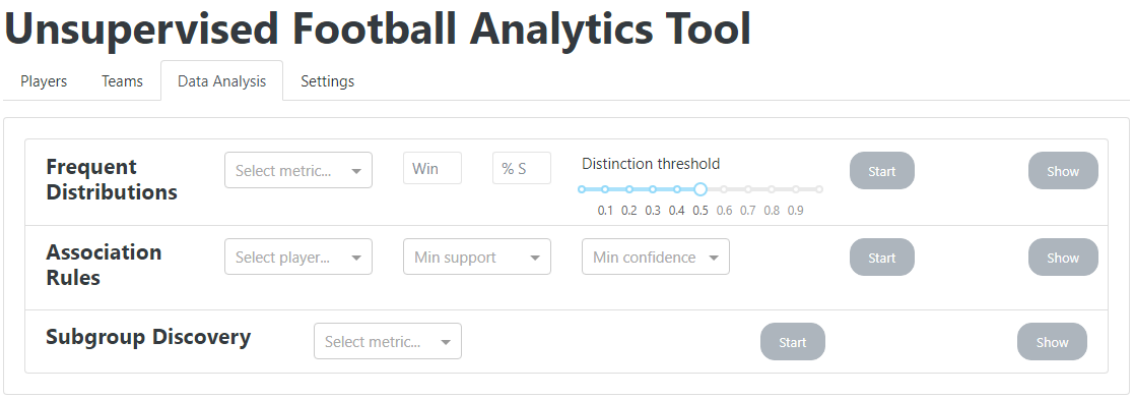


Figure 3.6: Image of the Data Analysis view

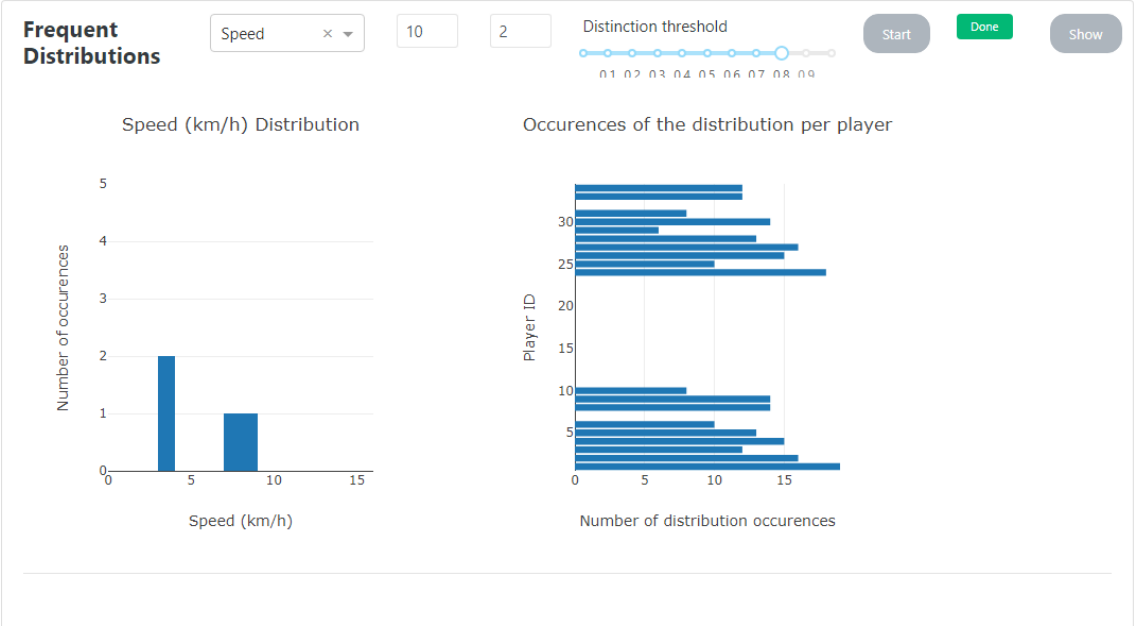


Figure 3.7: Example of the Frequent Distribution results of the tool

the player which we will analyze, another dropdown to select the minimum support for the frequent itemsets, and another dropdown to select the minimum confidence for the rules obtained. The third separator runs the Subgroup Discovery module. From left to right, there is a dropdown to select the target attribute, another dropdown to select the indication (higher or lower), and an input box to select the target value. An example visualization of the results can be seen in Figures 3.7, 3.8 and 3.9.

- Settings view: This is where the user uploads the match data file and chooses the parameters for the analysis, starting the pipeline. There are 2 main components in this view. The first is the file upload board (number 1 in Figure 3.10). In this component, the user inputs the values needed for the start of the loading phase, as described in Subsection 3.2.1. After the user selects/drags the file in the Upload File box, the folder is created as well as files containing the input values. The second component is the match analysis board (number 2 in Figure 3.10). The Select Match dropdown in this component allows selecting one of the match files that have already been uploaded to the tool. The user can input the size of the time window and then click on the Analyze button to start the processing phase. This phase is explained in Subsection 3.2.1. When a match has already been processed, a green label with the word "Done" will appear, meaning that it is already possible to analyze the data in the other 3 views. The Period switch in this component is used to switch the period of the displayed results.

Association Rules Player 2 | 10% | 50% | Start | Done | Show

Antecedents	Consequents	Support (%)	Confidence (%)
'prevPressure=0.0'	'afterPressure=0'	95	100
'prevAgility=4'	'afterPressure=0'	65	100
'prevPressure=0.0', 'prevAgility=4'	'afterPressure=0'	65	100
'prevSpeed=4'	'afterPressure=0'	60	100
'prevPressure=0.0', 'prevSpeed=4'	'afterPressure=0'	60	100
'prevSpeed=4', 'prevAgility=4'	'afterPressure=0'	40	100
'prevPressure=0.0', 'prevSpeed=4', 'prevAgility=4'	'afterPressure=0'	40	100
'prevStamina=4'	'afterPressure=0'	35	100
'prevPressure=0.0', 'prevStamina=4'	'afterPressure=0'	35	100
'prevSpeed=4', 'prevStamina=4'	'afterPressure=0'	35	100
'prevPressure=0.0', 'prevSpeed=4', 'prevStamina=4'	'afterPressure=0'	35	100
'prevStamina=3'	'afterPressure=0'	30	100
'prevStamina=3', 'prevAgility=4'	'afterPressure=0'	30	100
'prevPressure=0.0', 'prevStamina=3'	'afterPressure=0'	30	100
'prevPressure=0.0', 'prevStamina=3', 'prevAgility=4'	'afterPressure=0'	30	100
'prevSpeed=3'	'afterPressure=0'	25	100
'prevStamina=2'	'afterPressure=0'	25	100

Figure 3.8: Example of the Association Rules results of the tool

Subgroup Discovery Max Speed | >= | Start | Done | Show

Subgroup Description (speed>=3)	Score
stamina=4	150.9
stamina=1	112.98
stamina=1 AND pressure=0	107.35
avg_speed>=12.48	100.28
Def: [0:1[AND stamina=4	98.72
avg_speed>=12.48 AND stamina=4	91.26
avg_speed<4.71	90.96
stamina=4 AND pressure=0	90.89
avg_speed<4.71 AND pressure=0	85.52
stamina=4 AND Right: [0:1[	83.29

Figure 3.9: Example of the Subgroup Discovery results of the tool

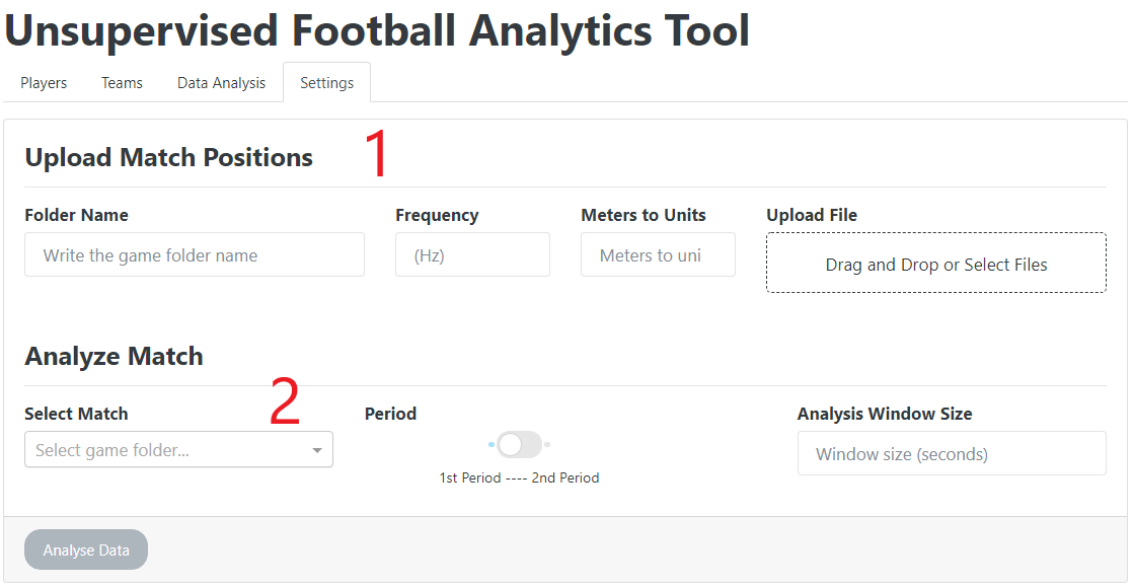


Figure 3.10: Image of the Settings view. 1 - File Upload board; 2 - Match Analysis board

Chapter 4

Results and Experimental Setup

On the start of this chapter, the datasets that were used for the experiments are described. It is given a general description of what they contain and how they were adapted to be used in the experiments. Afterwards, we present the results of the frequent distributions method, as well as its combination with association rules. In the end, we show the results of the tool itself. We compare the results of the scores obtained with the results of the real matches, and then we present and analyse the results obtained in the Association Rules and Subgroup Discovery modules.

4.1 Datasets

For the testing of the UnFOOT tool, seven different matches were used. Six of them were from Source A and one was from Source B.

4.1.1 Source A Datasets

The datasets from Source A represent 6 competitive matches of Team A versus other teams. The datasets were obtained with the Amisco system[1], and come in XML format. They include information about the match, the teams and each half of the match. After analyzing the dataset, we could infer the following aspects:

- The players' positions are measured 10 times per second (10 Hz).
- The players' x and y coordinates are in decimetres. x represents the width and y represents the length.
- The center of the field is $(x,y) = (0,0)$, and the approximate range of coordinates inside the field are $x = [-525,525]$ and $y = [-340,340]$.
- The team of the player can be inferred by his id (NumAmisco).
- The dataset also includes the positions of the ball and referees.

[1] https://www.stats.com/

An example of the dataset is shown in Appendix B (Listing B.1).

To parse the Source A XML files, a Python script was created to read the XML file and to store the data in the format seen in Table 3.1. This program consists of using an XML reader module to read the positions player by player.

4.1.2 Source B Dataset

The dataset from Source B represents one friendly match. It comes in the Electronic Performance and Tracking Systems (EPTS) Standard Data Format and comes in two files: a TXT file with the position measurements from the match, and an XML file with the configuration of the measurements. After analyzing the dataset, we could infer the following aspects:

- The players' positions are measured 25 times per second (25 Hz).
- The players' coordinates come in the format (x, y, z, v). x represents the length, y represents the width, z represents the height and v represents the speed. x, y, and z are in meters, and v is in m/s.
- The center of the field is $(x, y) = (34.0, 52.5)$ The approximate range of coordinates inside the field is $x = [0, 68]$ and $y = [0, 105]$. (z and v were ignored)
- While a player is not playing, his x and y values are 0. This applies before and after substitutions.
- When the tracking system could not detect the player, his x and y values were also 0.
- The recordings have some big jumps between consecutive measurements, which could not be humanly possible. This was due to the tracking system occasionally mistaking one player with another player/with the referee.
- The tracking system had difficulties tracking the goalkeepers.
- The dataset also includes the positions of the ball.
- The data, in general, was noisy.

An example of the configuration XML is shown in Listing B.2, and an example of the position measurements file is shown in Table B.1. The listing and the table can be found in Appendix B.

While the substitute players are not playing, their x and y values are 0. The UnFOOT tool interprets this as the substitute players being in the position (x, y) = (0, 0). This leads to multiple mistakes in the end results. For example, the substitute players positioning would not be correct, as well as the total playing time. This would also happen with the starting lineup players who were substituted later in the game. To solve this issue, we remove the leading and trailing (0, 0) positions of every player. To address the problem where players are temporarily not detected, we chose to reconstruct the missing values using linear interpolation. After solving these two problems, the Source B dataset is ready to be input in the tool.

4.1.3 Electricity Dataset

For the frequent distributions + association rule mining analysis, we also used the well-known electricity dataset[2] to test if this approach would work outside the sports context. Electricity dataset has the following aspects:

- Includes electricity data from Australian New South Wales and Victoria states from 1996 to 1998.
- The electricity records are measured every 30 minutes.
- Includes the normalized electricity price and demand for both states.

The electricity dataset needed to be transformed before being used in the experiments. The program expects that the data has a player_id. So, we gave to each state an id and named it "player_id" (New South Wales = 1, Victoria = 2). Also, the program expects to have consecutive numbered timestamps. So, we transformed the dataset so that each record refers to only one state. This means that each original dataset record was split in two: one regarding New South Wales state, and the other regarding the Victoria state. In the end of the transformation, the dataset had the format seen in Table 4.1.

Table 4.1: Example of the electricity dataset records after the transformation

date	day	period	price	demand	transfer	class	player_id	timestamp
0.00	2	0.00	0.056443	0.439155	0.414912	UP	1	0

4.2 Frequent Distributions Analysis

Recalling the definition of a profile found in Section 3.1, a profile is represented as a distribution of the values of a given attribute during a time interval. We used the Frequent Distributions method combined with association rule mining in order to obtain speed profiles of players and relationships between them. We could have also searched for profiles of distance covered, but due to time restrictions this was not possible to do in this project. Also, experiments were made with the Electricity dataset to obtain profiles of the electricity prices. This was done to verify if the method works in contexts other than football. A jupyter notebook was made to combine the two methods and display the results. Several experiments were made with different datasets and parameters. As explained in Chapter 3, *wsize* is the size of the time windows, and *threshold* is the distance threshold for two distributions to be considered distinct. Some preliminary experiments were made to find a value of *threshold* that produced clearly distinguishable profiles.

- Experiment 1: Source A - Match 1 dataset; $wsize = 50$; $threshold = 0.7$
- Experiment 2: Source A - Match 1 dataset; $wsize = 100$; $threshold = 0.7$

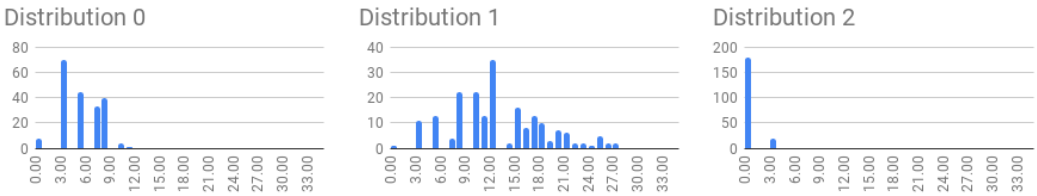
[2] https://moa.cms.waikato.ac.nz/datasets/

- Experiment 3: Source A - Match 1 dataset; $wsize = 100$; $threshold = 0.8$
- Experiment 4: Source A - Match 1 dataset; $wsize = 100$; $threshold = 0.9$
- Experiment 5: Source A - Match 1 dataset; $wsize = 200$; $threshold = 0.7$
- Experiment 6: SciSport dataset; $wsize = 100$; $threshold = 0.8$
- Experiment 7: SciSport dataset; $wsize = 250$; $threshold = 0.8$
- Experiment 8: SciSport dataset; $wsize = 500$; $threshold = 0.8$
- Experiment 9: Electricity dataset; $wsize = 144$; $threshold = 0.8$
- Experiment 10: Source A - Match 1 dataset; $wsize = 600$; $threshold = 0.6$
- Experiment 11: Electricity dataset; $wsize = 48$; $threshold = 0.8$
- Experiment 12: Electricity dataset; $wsize = 96$; $threshold = 0.8$

The results obtained in the experiments are shown in Appendix A. All rules which have $confidence < 0.5$ were omitted due to irrelevancy.

An aspect that is common to all experiments is that the smaller the *wsize* value is, the more profiles are found. This can be explained by the fact that having smaller windows implies that more distributions will be observed since each distribution will include fewer samples from the dataset. Fewer samples lead to more variability between the distributions observed, which in the end translates into finding more profiles.

From the results of the Source A dataset experiments (1, 2, 3, 4, 5, 10), we can observe that, in general, there are 3 main profiles. This is more evident in experiments 4, 5 and 10. (Figure 4.1).

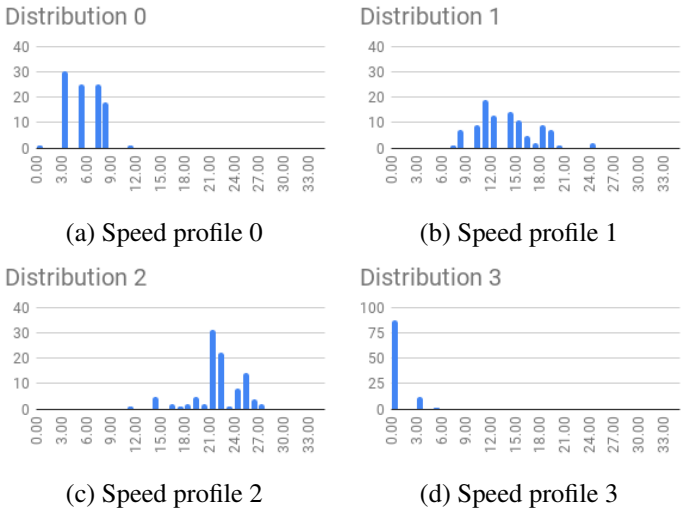


(a) Speed profile 0 (b) Speed profile 1 (c) Speed profile 2

Figure 4.1: Speed profiles of players obtained in Experiment 5: Source A

There is one profile for standing still/walking (Figure 4.1c), a second profile for slow running (Figure 4.1a) and third profile for bursts of speed (Figure 4.1b). This third profile has a lot more variability than the others since the range of speeds observed goes from 0 to 22km/h. This can mean that, in this profile, the player is not always running but increases and decreases his speed considerably.

In Experiments 2 and 3, four profiles were found (Figure 4.2).



(a) Speed profile 0 (b) Speed profile 1

(c) Speed profile 2 (d) Speed profile 3

Figure 4.2: Speed profiles of players obtained in Experiment 3: Source A

Similarly to Experiments 4, 5 and 10, there is one profile for standing still/walking and a profile for slow running (Figures 4.2d and 4.2a). The other two profiles are for running and sprinting (Figures 4.2b and 4.2c).

Regarding the relationships between profiles, it was found that is common for players to switch from the running profile to the slow running one. In Experiment 2, for 9 players it happened more than 50% of the times when they were in the running profile; in Experiment 3, for 2 players; in Experiment 5, for 14 players; and in Experiment 10, for every player. The rules are shown in Table 4.2, numbers 1, 2, 3 and 4. Also, it was found that was common for players to switch from the standing still/walking to the slow running profile. In Experiment 2, for 5 players it happened more than 50% of the times when they were in the standing still/walking profile; in Experiment 5, for 3 players; and in Experiment 10, for 15 players. The rules are shown in Table 4.2, numbers 5, 6 and 7. It was also found that was common for players to keep in the slow running profile. In Experiment 2, for 12 players it happened more than 50% of the times when they were in the slow running profile; in Experiment 3, for 2 players; in Experiment 5, for 12 players; in Experiment 10, for all players. The rules are shown in Table 4.2, numbers 8, 9, 10 and 11.

This reveals that, in the Source A dataset, players have a tendency to keep in the slow running profile or to return to it after being in a different profile.

In the Source B data, there was a big number of profiles found. Experiment 6 was found 102 profiles, in Experiment 7 were found 22, and in Experiment 8 were found 13. Most profiles in these 3 experiments involved distributions with the same unique value (assuming a precision of 1 decimal digit). Due to the unrealistic nature of the results, these were discarded. One of the reasons considered for the existence of the unrealistic profiles was the fact that some players' records were very noisy. These records had to suffer a substantial amount of filtering and transformations, which could have lead to poor results.

In the Electricity dataset, Experiments 11 and 12 had similar results to Source B: there was a

Table 4.2: Most important rules found in Match 2: Source A

rule_id	antecedent	consequent	support	confidence	number of players	experiment
1	prevDist=1	afterDist=0	3-16%	50-58%	9	Experiment 2
2	prevDist=1	afterDist=0	3%,12%	50%	2	Experiment 3
3	prevDist=1	afterDist=0	17-20%	50-54%	14	Experiment 5
4	prevDist=1	afterDist=0	6-25%	50-73%	22	Experiment 10
5	prevDist=3	afterDist=0	6-20%	50-53%	5	Experiment 2
6	prevDist=2	afterDist=0	6%	51-52%	3	Experiment 5
7	prevDist=2	afterDist=0	1-17%	50-67%	3	Experiment 10
8	prevDist=0	afterDist=0	25-31%	51-56%	12	Experiment 2
9	prevDist=0	afterDist=0	25%	50%	2	Experiment 3
10	prevDist=0	afterDist=0	25-29%	50-54%	12	Experiment 5
11	prevDist=0	afterDist=0	25-47%	50-68%	22	Experiment 10

big number of profiles found, each with just the same unique value. However, for Experiment 9 were found only 7 profiles. This can mean that using bigger windows may allow obtaining more meaningful profiles. The profiles are shown in Figure 4.3.

The rules found were only relative to the Victoria state. The rules state that, in more than 60% of the times that Victoria state was in the profile 1 or 3, it stayed on that profile. These rules are shown in Table 4.3. Those two profiles are the ones which involve the cheapest prices. This means that, when is observed that the city has very cheap electricity prices for 2 days (wsize = 144), it is likely to keep the same for the next 2 days.

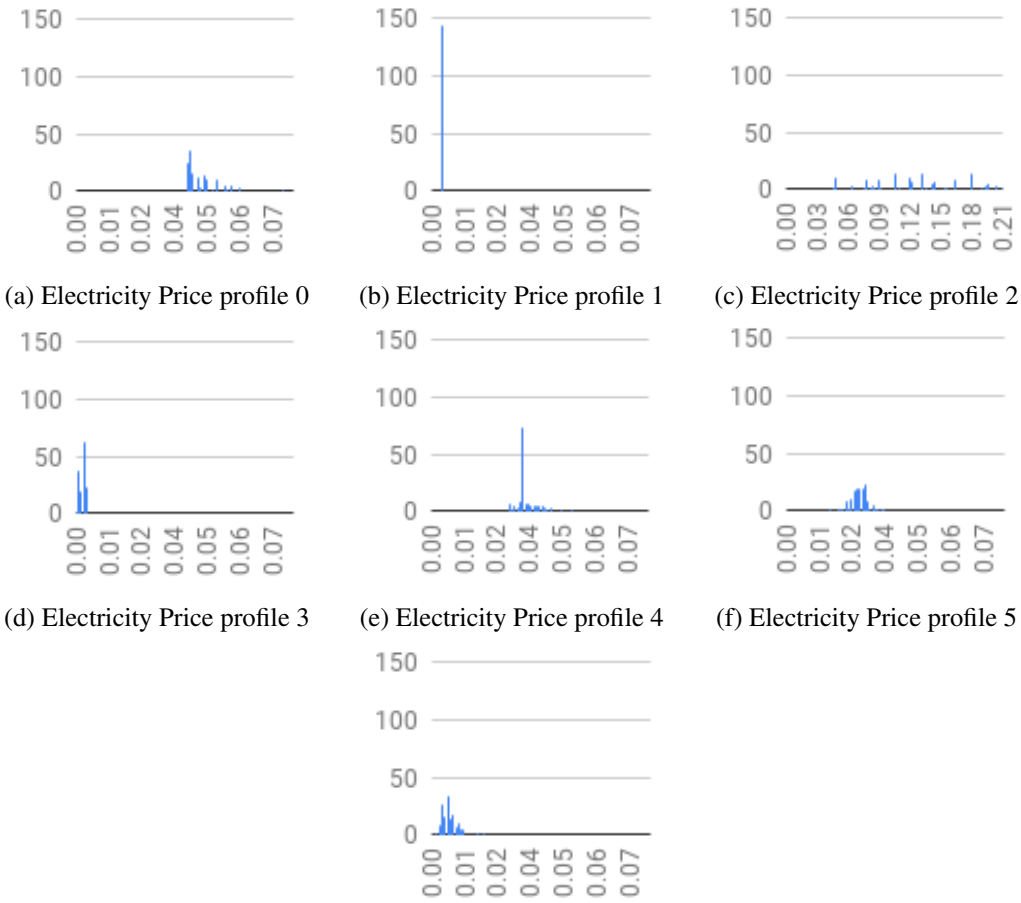
Table 4.3: Association rules found in the Electricity dataset relative to the Victoria state: Experiment 9

rule_id	antecedent	consequent	support	confidence
1	prevDist=1	afterDist=1	27%	64%
2	prevDist=3	afterDist=3	24%	62%

4.3 UnFOOT Analysis

To analyze the UnFOOT results, we used the datasets from Source A. Due to external sources, we could know more details about the match, such as the player of the match, which team won and what are the usual roles of each player. We used this external information to validate our results. However, some details cannot be shown in order to preserve the privacy of the players involved.

According to some metrics obtained (Table 4.4), the best player of the match was usually found on the top three players of the winning team. In two cases, they even had the best score overall. We note that the overall score was not originally designed to predict the best player of the match. Still, we used it to validate the scoring function. We should also note that this scoring function can only reasonably assess the quality of players, which are no goalkeepers. This is seen in Game 5, where the best player was actually a goalkeeper.



(a) Electricity Price profile 0 (b) Electricity Price profile 1 (c) Electricity Price profile 2

(d) Electricity Price profile 3 (e) Electricity Price profile 4 (f) Electricity Price profile 5

(g) Electricity Price profile 6

Figure 4.3: Profiles of the electricity prices. Note that distribution 2 scale is different from the others.

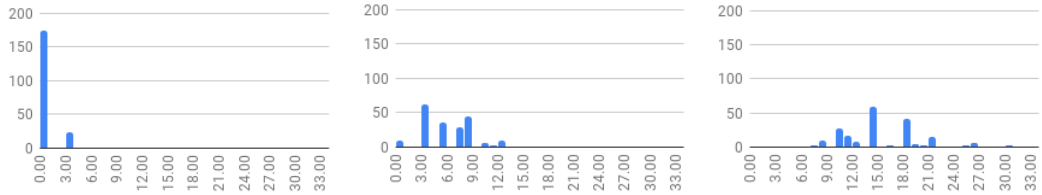
Relatively to the positioning module results, which we can see in Table 4.5, the positioning labels given by the module matched the players' roles around 67% of the times. Match 3 was ignored since there was a problem with the data when gathering these results. There are two possible reasons for the mismatch between the label and the role of the players. The first reason could be that the players with a mismatch played very differently from their initial role. For example, a right midfielder could have played more in the offensive because the left defender from the other team gave him space. Because of that, that player would have been labeled as a Right Winger by the tool, instead of a Midfielder. The second reason could be because of the way that the positioning of the player is calculated, which is explained in Subsection 3.2.1. Since the players' positions are divided into 3 sections with the same positions count, it could happen that two very similar positions were assigned to different sections. This is better illustrated in an example: if the total number of positions was 33, then we would have 11 positions labeled as Defense, 11 as Midfield and 11 as Attack. So, if we had 12 positions that were clearly should be labeled as defensive, only the first 11 would be labeled as that.

Table 4.4: Comparison between the real match results and the results of the tool

Match	Winner	Team A score	Team B score	Rank of best player
1	A	758	778	3rd of Team A
2	A	814	811	1st overall
3	A	795	805	3rd of Team A
4	B	832	855	3rd of Team B
5	A	813	796	Last overall
6	A	816	819	1st overall

Relatively to the team scores, which we can also see in Table 4.4, we cannot infer from the team global scores who was the winning team. In only 3 of the 6 matches analyzed the winning team was the one with the highest score. The team global score reflects the sum of the individual scores of the players, as explained in Subsection 3.2.3. From this, we can deduce that the sum of individual performances may not be enough to evaluate the performance of the whole team. We also have to consider that the player individual scores do not have into account aspects such as a shot or pass accuracy, or even communication and teamwork. This could mean that positional data analysis is not enough to infer which team won and that event data analysis is needed in this case.

The results of the frequent distributions module obtained in UnFOOT are consistent with the ones obtained on the experiments done in Section 4.2. In all matches, when searching for speed profiles, we obtained 3 main profiles, except for Match 2, where only 2 profiles where found. A plot of the profiles is shown in Figure 4.4. For the parameters values, we used $wsize = 200$ and $threshold = 0.8$. The first main profile reflects walking speed, the second one reflects slow running speed, and the third one reflects running/sprinting speed.



(a) Speed profile 0 (b) Speed profile 1 (c) Speed profile 2

Figure 4.4: Speed profiles of the players in Match 2.

The first profile was not found in Match 2. A possible interpretation for this is that Match 2 was a more intense game, there was less frequent to register low speeds for a long time.

4.3.1 Association Rules

Regarding the association rules module results, the following aspects need to be mentioned:

- The Speed, Stamina, and Agility values vary from 1 to 4, each one representing a quartile, from the lowest to the highest score (Section 3.2).
- The Pressure values vary between 1 and 10.

Table 4.5: Number of players correctly classified by the position module in each match of Source A.

Match	Correct
1	16 out of 22
2	14 out of 22
3	0 out of 22
4	14 out of 22
5	15 out of 22
6	15 out of 22

- The matches analyzed where Match 1 and 2 from Source A.

In both matches we observed that most rules had Pressure=0 in the consequents. Given the high frequency of items with Pressure=0 in the itemsets, it would be very common to find the item with Pressure=0 in the consequents, no matter what the other items were. Nevertheless, we can infer that players spend most of the match away from pressure situations.

In Match 2, one of the rules indicated that, players tend to keep non-intermediate performance (high or low) throughout the game. This happened for at least 19 players out of 22. At least in 10% of Match 2, players had Speed/Stamina=1, which kept the same at least 50% of the times. This is shown in Table 4.6, in rules 1, 2, 3 and 4. The exact value for the percentages depends on the player. The goalkeepers have higher support/confidence in the low performance rules (rules 1 and 2) but the high performance rules (rules 3 and 4) do not apply to them. One of the best rules in Match 2 indicated that one striker of Team B (player 33) was subject to a lot of pressure during the match. During 13% of the match, this player had an intermediate pressure score, which was followed by a high pressure score in 81% of the time. This is shown in Table 4.6, in rule 7.

In Match 1, one rule showed that player 28 frequently revealed very high intensity periods while not in pressure situations. In at least 22% of the match, player 28 had Speed/Stamina=4 and Pressure=0, followed by Pressure=0 again in at least 88% of the times. The rules are shown in Table 4.6, in rules 5 and 6. Player 28 is a Right Back, which is a position where usually the player is more isolated than midfielders and forwards and has to run along the sides of the field. This could be a possible explanation for the finding of this rule.

Table 4.6: Best association rules of Match 1 and 2

rule_id	antecedent	consequent	support	confidence	player_id	match
1	Speed=1	Speed=1	10-50%	50-79%	multiple	Match 2
2	Stamina=1	Stamina=1	10-61%	53-86%	multiple	Match 2
3	Speed=4	Speed=4	11-23%	52-66%	multiple	Match 2
4	Stamina=4	Stamina=4	12-27%	54-72%	multiple	Match 2
5	Pressure=0, Speed=4	Pressure=0	24%	89%	28	Match 1
6	Pressure=0, Stamina=4	Pressure=0	22%	88%	28	Match 1
7	Pressure=6	Pressure=9	13%	81%	33	Match 2

4.3.2 Subgroup Discovery

Regarding the subgroup discovery module results, some additional aspects must be mentioned:

- The aspects from the Association Rules results section (Subsection 4.3.1) still maintain.
- With exception of the Agility score, that has an average value of approximately 30 in Match 1, and 27 in Match 2.
- Def, Med and Atk values vary between 0 and 10.

In both matches, two subgroups indicate that, when players achieved higher speed scores, they usually revealed either a higher stamina score, or a lower stamina score. The subgroups that indicate this are subgroups 1 and 2 of Table 4.7. It were also found two subgroups which indicate that high pressure was observed when players were playing in the defense. This is indicated by subgroups 6 and 7 of Table 4.7. However, one particular subgroup also indicates that high pressure score was observed in situations where players were not in the defense and where the average speed was low (subgroup 8 and 9 of Table 4.7). A possible explanation for this is that these subgroups may correspond to strikers of both teams. In the tool, it was verified in the Player view that the strikers have a high pressure score. If we assume that strikers wait for the ball near the opposite team's defense line, and that they do not run to save the energy for counterattacks, then the above explanation could be plausible.

In match 2, three subgroups indicate that players showed high agility scores when they were playing in the offensive with a high speed score. This is indicated by subgroups 3, 4 and 5 of Table 4.7. All of these subgroups include the highest speed quartile (Speed=4) and show that a high agility score was usually observed when players were playing in the attack (Def=0, Med=0, Atk=10).

Table 4.7: Subgroups found in Match 1 and 2 with different targets

subgroup_id	subgroup	target	match
1	Stamina=1	Speed≥3	both
2	Stamina=4	Speed≥3	both
3	Speed=4 AND Med=0	Agility≥80	Match 2
4	Speed=4 AND Def=0	Agility≥80	Match 2
5	Speed=4 AND Atk=10	Agility≥80	Match 2
6	Atk=0 AND Med=0	Pressure≥9	both
7	Def=10 AND Atk=0	Pressure≥9	both
8	AverageSpeed<2.82m/s AND Def=0	Pressure≥9	Match 1
9	AverageSpeed<3.12m/s AND Def=0	Pressure≥9	Match 2

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This project had the objective of providing a data mining tool to aid sports trainers, sports analysts and data scientists in the visualization and analysis of player's spatiotemporal data from a match. This was accomplished through the UnFOOT tool.

One of the objectives was giving insights on the performance of football and hockey players. UnFOOT was built to use football data, but can easily be extended to use hockey data. Also, feature engineering was supposed to be applied on the data. This was accomplished, since we identified and extracted features from the data, such as the players' velocity, acceleration and distance covered. The extracted features allowed to obtain the performance metrics of the players. Another objective was implementing a data mining module, which was accomplished in the Data Analysis interface, which includes subgroup discovery and association rules mining. Additionally, in the data analysis interface, a new method called Frequent Distributions was developed for obtaining players profiles regarding performance metrics such as speed and stamina. Finally, the tool should provide a data visualization module. This was accomplished through two different interfaces: one to visualize player performance, and another to visualize team performance. In these interfaces the user could analyse and compare players' and teams' performance.

The player scoring function could make a rough estimate of the players' performance, except for the goalkeepers, but the team scoring function was not able to describe the teams' performance. Also, the positioning module could infer most of the players' positioning roles in the team.

The aim of the project was using spatiotemporal data from players, so no analysis of ball data or match events data is done.

5.2 Future Work

Solving some remaining bugs in the tool should be addressed in the future. In the Data Analysis interface, a possible improvement is the implementation of other data mining algorithms, which

can be useful in the analysis of player spatiotemporal data. Regarding the positioning module, further improvements can be made to increase the number of times that the tool can infer the player's positioning role in the team, or also suggest multiple possible roles. Automatic recognition of teams' formation can be a possible addition to this module as well. Regarding the team scores, a new way of calculating the team score should be found, since the current one cannot evaluate well the performance of the teams. The player scores and metric calculation could also be improved in the future, in order to better evaluate goalkeepers. Also, improving the treatment of the noisy data should be considered, to get better results in datasets similar to Source B. Future research can involve further study in the Frequent Distributions method.

References

[AFGY02] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 429–435. ACM, 2002.

[AG95] HELMUT ALT and MICHAEL GODAU. Computing the frÉchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 05(01n02):75–91, 1995.

[AKW04] Helmut Alt, Christian Knauer, and Carola Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38(1):45–58, Jan 2004.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499, 1994.

[BBG+11] KEVIN BUCHIN, MAIKE BUCHIN, JOACHIM GUDMUNDSSON, MAARTEN LÖFFLER, and JUN LUO. Detecting commuting patterns by clustering subtrajectories. *International Journal of Computational Geometry & Applications*, 21(03):253–282, 2011.

[BD13] P.J. Brockwell and R.A. Davis. *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer New York, 2013.

[Bis16] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York, 2016.

[BLC+14a] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews. Identifying team style in soccer using formations learned from spatiotemporal tracking data. In *2014 IEEE International Conference on Data Mining Workshop*, pages 9–14, Dec 2014.

[BLC+14b] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *2014 IEEE International Conference on Data Mining*, pages 725–730, Dec 2014.

[Con71] W.J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, 1971.

[DGH+18] J. Degele, A. Gorr, K. Haas, D. Kormann, S. Krauss, P. Lipinski, M. Tenbih, C. Koppenhoefer, J. Fauser, and D. Hertweck. Identifying e-scooter sharing customer segments using clustering. In *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–8, June 2018.

[dLMK18] Arie-Willem de Leeuw, Laurentius A. Meerhoff, and Arno Knobbe. Effects of pacing properties on performance in long-distance running. *Big Data*, 6(4):248–261, 2018. PMID: 30421990.

[FS05] Akira Fujimura and Kokichi Sugihara. Geometric analysis and quantitative evaluation of sport teamwork. *Systems and Computers in Japan*, 36(6):49–58, 2005.

[GH17] Joachim Gudmundsson and Michael Horton. Spatio-temporal analysis of team sports. *ACM Comput. Surv.*, 50(2):22:1–22:34, April 2017.

[GW14] Joachim Gudmundsson and Thomas Wolle. Football analysis using spatio-temporal tools. *Computers, Environment and Urban Systems*, 47:16 – 27, 2014. Progress in Movement Analysis – Experiences with Real Data.

[Hel16] Sumyea Helal. Subgroup discovery algorithms: A survey and empirical evaluation. *Journal of Computer Science and Technology*, 31(3):561–576, May 2016.

[HGER15a] Keith Henderson, Brian Gallagher, and Tina Eliassi-Rad. Ep-means: An efficient nonparametric clustering of empirical probability distributions. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 893–900, New York, NY, USA, 2015. ACM.

[HGER15b] Keith Henderson, Brian Gallagher, and Tina Eliassi-Rad. Ep-means: An efficient nonparametric clustering of empirical probability distributions. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 893–900, New York, NY, USA, 2015. ACM.

[HLW13] Xin Huang, Jun Luo, and Xin Wang. Finding frequent sub-trajectories with time constraints. In *Proceedings of the 2Nd ACM SIGKDD International Workshop on Urban Computing*, UrbComp '13, pages 13:1–13:8, New York, NY, USA, 2013. ACM.

[JAP06] Alípio M. Jorge, Paulo J. Azevedo, and Fernando Pereira. Distribution rules with numeric attributes of interest. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Knowledge Discovery in Databases: PKDD 2006*, pages 247–258, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[LC18] Jie Lin and Robert G. Cromley. Inferring the home locations of twitter users based on the spatiotemporal clustering of twitter data. *Transactions in GIS*, 22(1):82–97, 2018.

[MdLGK] Laurentius A Meerhoff, Arie-Willem de Leeuw, Floris R Goes, and Arno Knobbe. Mining soccer data: Subgroup discovery of tactics from spatio-temporal data.

[MPS18] Ferdinando Di Martino, Witold Pedrycz, and Salvatore Sessa. Spatiotemporal extended fuzzy c-means clustering algorithm for hotspots detection and prediction. *Fuzzy Sets and Systems*, 340:109 – 126, 2018. Theme: Clustering.

[NTO18] S. Naghdi, C. Tjhai, and K. O'Keefe. Assessing a uwb rtls as a means for rapid wlan radio map generation. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–5, Sep. 2018.

[RER99] S. J. Roberts, R. Everson, and I. Rezek. Minimum entropy data partitioning. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 844–849 vol.2, Sep. 1999.

[RTG98] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66, Jan 1998.

[SLT+18] M.-W. Su, W.-C. Lin, C.-H. Tsai, B.-L. Chiang, Y.-H. Yang, Y.-T. Lin, L.-C. Wang, J.-H. Lee, C.-C. Chou, Y.-F. Wu, Y.-L. Yeh, and Y. L. Lee. Childhood asthma clusters reveal neutrophil-predominant phenotype with distinct gene expression. *Allergy*, 73(10):2024–2032, 2018.

[SRP17] Gabriel J Sanders, Brad Roll, and Corey A Peacock. Maximum distance and high-speed distance demands by position in ncaa division i collegiate football games. *The Journal of Strength & Conditioning Research*, 31(10):2728–2733, 2017.

[TH00] T. Taki and J. Hasegawa. Visualization of dominant region in team games and its application to teamwork analysis. In *Proceedings Computer Graphics International 2000*, pages 227–235, June 2000.

[TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

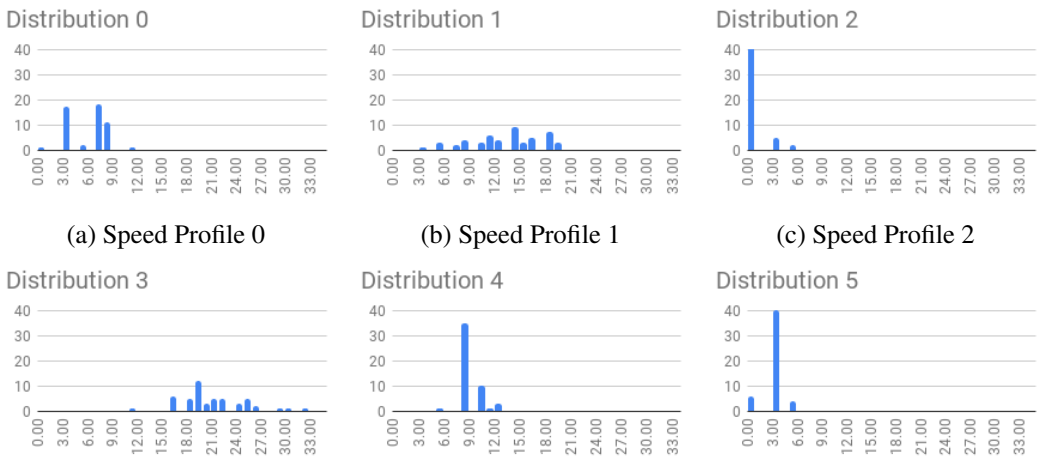
[VBT09] Marcos R. Vieira, Petko Bakalov, and Vassilis J. Tsotras. On-line discovery of flock patterns in spatio-temporal data. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 286–295, New York, NY, USA, 2009. ACM.

[YLC+14] Y. Yue, P. Lucey, P. Carr, A. Bialkowski, and I. Matthews. Learning fine-grained spatial models for dynamic sports play prediction. In *2014 IEEE International Conference on Data Mining*, pages 670–679, Dec 2014.

REFERENCES

Appendix A

Frequent Itemsets + Association Rules experiments



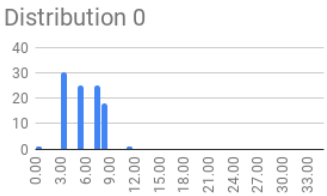
(a) Speed Profile 0 (b) Speed Profile 1 (c) Speed Profile 2

(d) Speed Profile 3 (e) Speed Profile 4 (f) Speed Profile 5

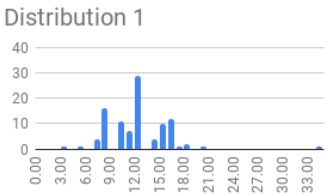
Figure A.1: Speed profiles of players in Experiment 1

Table A.1: Association rules found in Experiment 1

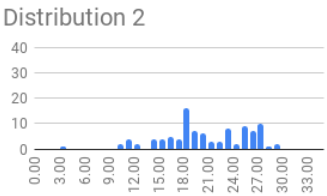
player_id	antecedents	consequents	support	confidence
1	prevProfile=1	afterProfile=0	0.02	0.42



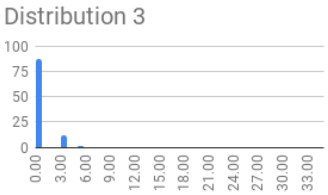
(a) Speed Profile 0



(b) Speed Profile 1



(c) Speed Profile 2

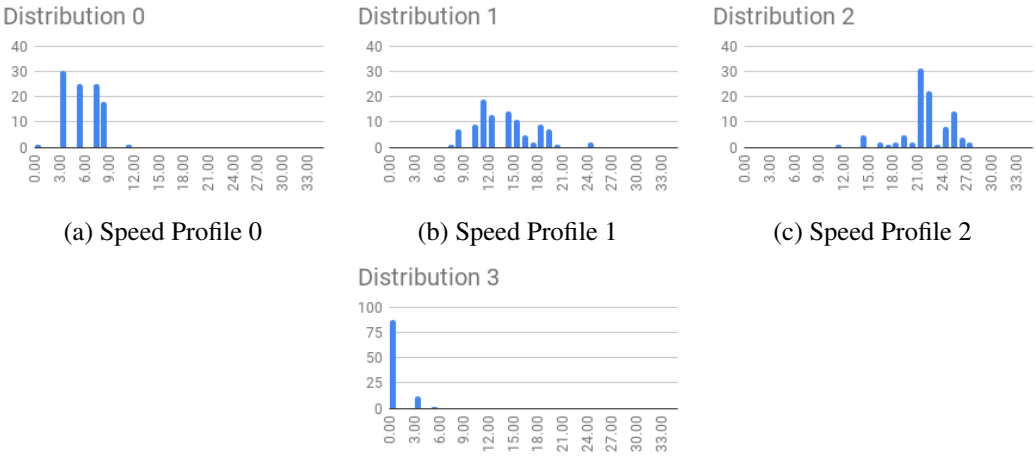


(d) Speed Profile 3

Figure A.2: Speed profiles of players in Experiment 2

Table A.2: Association rules found in Experiment 2

player_id	antecedents	consequents	support	confidence
1	prevDist=0	afterDist=0	0.29	0.54
1	prevDist=1	afterDist=0	0.04	0.58
1	prevDist=3	afterDist=0	0.20	0.52
3	prevDist=0	afterDist=0	0.25	0.50
4	prevDist=0	afterDist=0	0.26	0.51
4	prevDist=1	afterDist=0	0.14	0.51
4	prevDist=3	afterDist=0	0.07	0.51
5	prevDist=0	afterDist=0	0.25	0.51
9	prevDist=0	afterDist=0	0.27	0.53
9	prevDist=1	afterDist=0	0.14	0.52
11	prevDist=0	afterDist=0	0.25	0.51
11	prevDist=1	afterDist=0	0.14	0.51
24	prevDist=0	afterDist=0	0.25	0.51
24	prevDist=1	afterDist=0	0.03	0.56
25	prevDist=0	afterDist=0	0.30	0.55
25	prevDist=1	afterDist=0	0.14	0.54
25	prevDist=3	afterDist=0	0.07	0.51
26	prevDist=0	afterDist=0	0.26	0.52
26	prevDist=1	afterDist=0	0.14	0.51
27	prevDist=3	afterDist=0	0.05	0.50
30	prevDist=0	afterDist=0	0.26	0.52
30	prevDist=1	afterDist=0	0.16	0.51
33	prevDist=0	afterDist=0	0.24	0.50
34	prevDist=0	afterDist=0	0.25	0.51
34	prevDist=1	afterDist=0	0.14	0.50
34	prevDist=3	afterDist=0	0.07	0.50



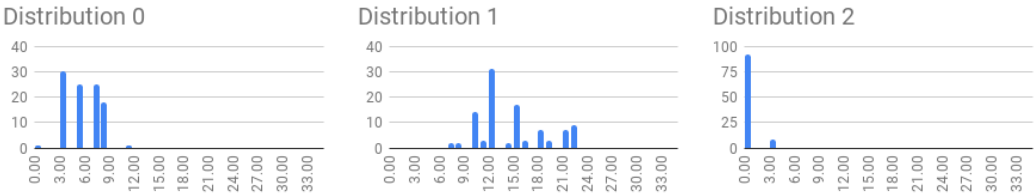
(a) Speed Profile 0 (b) Speed Profile 1 (c) Speed Profile 2

(d) Speed Profile 3

Figure A.3: Speed profiles of players in Experiment 3

Table A.3: Association rules found in Experiment 2

player_id	antecedents	consequents	support	confidence
1	prevDist=0	afterDist=0	0.24	0.5
1	prevDist=1	afterDist=0	0.03	0.5
25	prevDist=0	afterDist=0	0.25	0.5
25	prevDist=1	afterDist=0	0.11	0.5

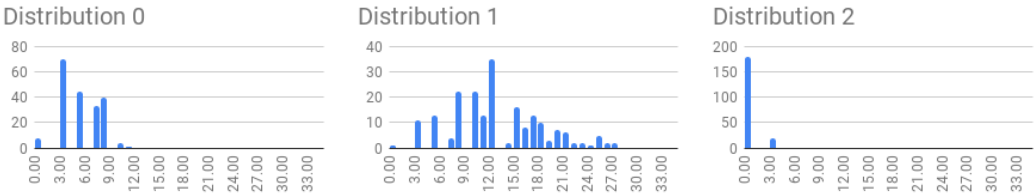


(a) Speed Profile 0 (b) Speed Profile 1 (c) Speed Profile 2

Figure A.4: Speed profiles of players in Experiment 4

Table A.4: Association rules found in Experiment 4

player_id	antecedents	consequents	support	confidence
-	-	-	-	-



(a) Speed Profile 0 (b) Speed Profile 1 (c) Speed Profile 2

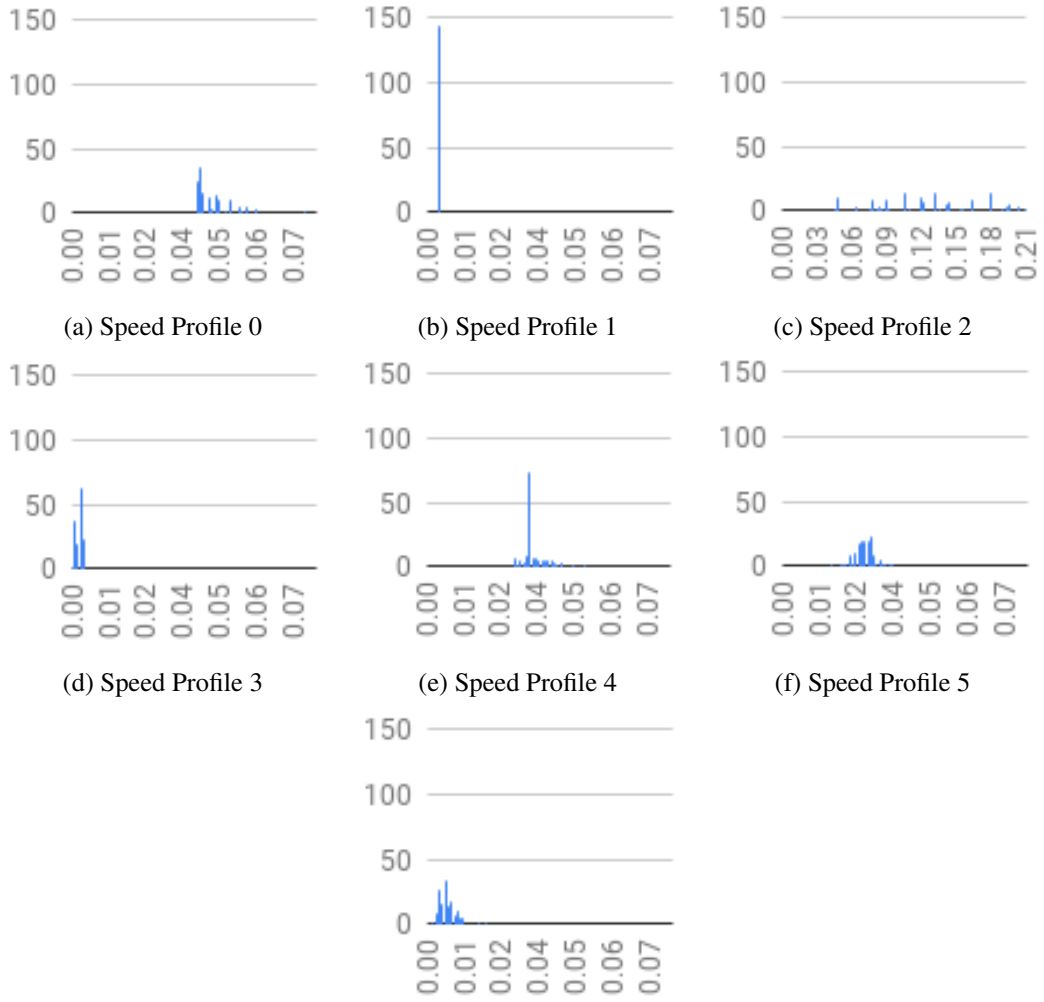
Figure A.5: Speed profiles of players in Experiment 5

Table A.5: Association rules found in Experiment 5

player_id	antecedents	consequents	support	confidence
1	prevDist=0	afterDist=0	0.25	0.51
2	prevDist=1	afterDist=0	0.18	0.50
4	prevDist=0	afterDist=0	0.29	0.54
4	prevDist=1	afterDist=0	0.19	0.54
4	prevDist=2	afterDist=0	0.06	0.52
5	prevDist=0	afterDist=0	0.25	0.50
5	prevDist=1	afterDist=0	0.17	0.50
6	prevDist=0	afterDist=0	0.25	0.51
6	prevDist=1	afterDist=0	0.19	0.52
7	prevDist=1	afterDist=0	0.20	0.50
9	prevDist=0	afterDist=0	0.26	0.52
9	prevDist=1	afterDist=0	0.19	0.52
11	prevDist=0	afterDist=0	0.26	0.51
11	prevDist=1	afterDist=0	0.18	0.52
25	prevDist=0	afterDist=0	0.29	0.54
25	prevDist=1	afterDist=0	0.19	0.54
25	prevDist=2	afterDist=0	0.06	0.52
27	prevDist=0	afterDist=0	0.26	0.51
27	prevDist=1	afterDist=0	0.20	0.51
29	prevDist=1	afterDist=0	0.20	0.51
30	prevDist=0	afterDist=0	0.28	0.53
30	prevDist=1	afterDist=0	0.20	0.54
31	prevDist=0	afterDist=0	0.26	0.51
31	prevDist=1	afterDist=0	0.20	0.51
33	prevDist=0	afterDist=0	0.26	0.51
33	prevDist=1	afterDist=0	0.19	0.51
34	prevDist=0	afterDist=0	0.25	0.50
34	prevDist=1	afterDist=0	0.19	0.50
34	prevDist=2	afterDist=0	0.06	0.51

Table A.6: Association rules found in Experiment 9

player_id	antecedents	consequents	support	confidence
2	afterDist=1	prevDist=1	0.27	0.64
2	prevDist=1	afterDist=1	0.27	0.64
2	afterDist=3	prevDist=3	0.24	0.61
2	prevDist=3	afterDist=3	0.24	0.62

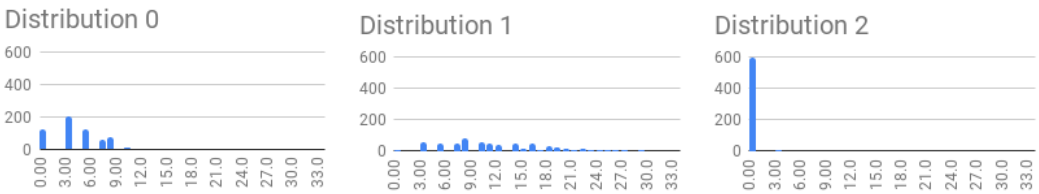


(a) Speed Profile 0 (b) Speed Profile 1 (c) Speed Profile 2

(d) Speed Profile 3 (e) Speed Profile 4 (f) Speed Profile 5

(g) Speed Profile 6

Figure A.6: Speed profiles of players in Experiment 9



(a) Speed Profile 0 (b) Speed Profile 1 (c) Speed Profile 2

Figure A.7: Speed profiles of players in Experiment 10

Table A.7: Association rules found in Experiment 10 for Team 1

player_id	antecedents	consequents	support	confidence
1	prevDist=0	afterDist=0	0.47	0.68
1	prevDist=1	afterDist=0	0.11	0.73
1	prevDist=2	afterDist=0	0.10	0.67
2	prevDist=0	afterDist=0	0.26	0.51
2	prevDist=1	afterDist=0	0.24	0.51
2	prevDist=2	afterDist=0	0.01	0.50
2	prevDist=2	afterDist=1	0.01	0.50
3	prevDist=0	afterDist=0	0.28	0.52
3	prevDist=1	afterDist=0	0.23	0.52
3	prevDist=2	afterDist=0	0.01	0.50
3	prevDist=2	afterDist=1	0.01	0.50
4	prevDist=0	afterDist=0	0.29	0.53
4	prevDist=1	afterDist=0	0.24	0.54
5	prevDist=0	afterDist=0	0.27	0.51
5	prevDist=1	afterDist=0	0.23	0.52
5	prevDist=2	afterDist=0	0.02	0.50
6	prevDist=0	afterDist=0	0.26	0.51
6	prevDist=1	afterDist=0	0.24	0.51
6	prevDist=2	afterDist=0	0.01	0.50
6	prevDist=2	afterDist=1	0.01	0.50
7	prevDist=0	afterDist=0	0.26	0.51
7	prevDist=1	afterDist=0	0.24	0.51
7	prevDist=2	afterDist=0	0.01	0.50
7	prevDist=2	afterDist=1	0.01	0.50
8	prevDist=0	afterDist=0	0.26	0.51
8	prevDist=1	afterDist=0	0.24	0.51
8	prevDist=2	afterDist=0	0.01	0.50
8	prevDist=2	afterDist=1	0.01	0.50
9	prevDist=0	afterDist=0	0.28	0.53
9	prevDist=1	afterDist=0	0.24	0.53
10	prevDist=0	afterDist=0	0.25	0.50
10	prevDist=1	afterDist=0	0.23	0.50
11	prevDist=0	afterDist=0	0.27	0.52
11	prevDist=1	afterDist=0	0.22	0.52
11	prevDist=2	afterDist=0	0.02	0.50

Table A.8: Association rules found in Experiment 10 for Team 2

player_id	antecedents	consequents	support	confidence
24	prevDist=0	afterDist=0	0.37	0.61
24	prevDist=1	afterDist=0	0.06	0.70
24	prevDist=2	afterDist=0	0.17	0.55
25	prevDist=0	afterDist=0	0.28	0.53
25	prevDist=1	afterDist=0	0.24	0.54
25	prevDist=2	afterDist=0	0.01	0.50
25	prevDist=2	afterDist=1	0.01	0.50
26	prevDist=0	afterDist=0	0.29	0.54
26	prevDist=1	afterDist=0	0.22	0.54
26	prevDist=2	afterDist=0	0.03	0.50
26	prevDist=2	afterDist=1	0.03	0.50
27	prevDist=0	afterDist=0	0.26	0.51
27	prevDist=1	afterDist=0	0.24	0.51
28	prevDist=0	afterDist=0	0.25	0.50
28	prevDist=1	afterDist=0	0.25	0.50
28	prevDist=0	afterDist=1	0.25	0.50
28	prevDist=1	afterDist=1	0.25	0.50
29	prevDist=0	afterDist=0	0.26	0.52
29	prevDist=1	afterDist=0	0.24	0.52
29	prevDist=2	afterDist=0	0.02	0.50
29	prevDist=2	afterDist=1	0.02	0.50
30	prevDist=0	afterDist=0	0.27	0.52
30	prevDist=1	afterDist=0	0.24	0.53
31	prevDist=0	afterDist=0	0.28	0.53
31	prevDist=1	afterDist=0	0.24	0.53
32	prevDist=0	afterDist=0	0.27	0.52
32	prevDist=1	afterDist=0	0.24	0.52
32	prevDist=2	afterDist=0	0.01	0.50
32	prevDist=2	afterDist=1	0.01	0.50
33	prevDist=0	afterDist=0	0.27	0.52
33	prevDist=1	afterDist=0	0.24	0.53
33	prevDist=2	afterDist=0	0.02	0.50
33	prevDist=2	afterDist=1	0.02	0.50
34	prevDist=0	afterDist=0	0.27	0.52
34	prevDist=1	afterDist=0	0.24	0.53
34	prevDist=2	afterDist=0	0.01	0.50
34	prevDist=2	afterDist=1	0.01	0.50

Appendix B

Dataset Examples

```
<MATCH_SHEET Competition="Source A Competition 1" Date="14/06/19" PitchCode="STADI"
     Pitch="Stadium 1" DurationTime="45" ExtraTimeDuration="15" PitchWidth="
    105.000000" PitchHeight="68.000000">
    <tTeam Id="1234" Name="Team FC" Color1="1231231">
        <PLAYER NumAmisco="123123" SecondName="Ronaldo" FirstName="Cristiano"
            ShirtNumber="7"/>
        <PLAYER ... />
        ...
    </tTeam>
    <tTeam Id="2345" Name="Team 2 FC" Color1="2342342">
        <PLAYER ... />
        <PLAYER ... />
        ...
    </tTeam>
    <tTeam Id="0" Name="" Color="16777215">
        <PLAYER ... />
        <PLAYER ... />
        ...
    </tTeam>
</MATCH_SHEET>
<TRAJECTORIES>
    <PERIOD Id="1">
        <PLAYER NumAmisco="0" StartTime="0" EndTime="28176">
            <Pos Times="0" X="0" Y="0"/>
            <Pos Times="1" X="-1" Y="-10"/>
            <Pos .../>
            ...
        </PLAYER>
        <PLAYER ...>
            ...
        </PLAYER>
        ...
    </PERIOD>
```

```
    <PERIOD Id="2">
        ...
    </PERIOD>
</TRAJECTORIES>
```

Listing B.1: Example of the Source A dataset. Some attributes and values have been ommited due to simplicity and data confidentiality.

```
<Metadata>
    <GlobalConfig>
        <FrameRate>25</FrameRate>
        <Encoding>UTF-8</Encoding>
        <ProviderGlobalParameters>
            <ProviderParameter>
                <Value>34.0,52.5</Value>
                <Description>X and Y coordinates of the centre of the pitch</
                    Description>
                <Name>pitchcentre</Name>
            </ProviderParameter>
        </ProviderGlobalParameters>
    </GlobalConfig>
    <Sessions>
        <Session id="0">
            <SessionType>Period</SessionType>
            <SessionName>1H</SessionName>
            <Start>2019-02-26T18:00:37.886+01:00</Start>
            <End>2019-02-26T18:45:18.325+01:00</End>
            <MatchParameters>
              <FieldSize>
                <Length>105.0</Length>
                <Width>68.0</Width>
              </FieldSize>
            </MatchParameters>
            <Location>Name of the location</Location>
            <ProviderSessionParameters>
              <ProviderParameter>
                <Value>8448</Value>
                <Description>Frameid for the start of this session</Description>
                <Name>frame</Name>
              </ProviderParameter>
            </ProviderSessionParameters>
        </Session>
        <Session id="1">
            <SessionType>Period</SessionType>
            <SessionName>2H</SessionName>
            (...)
        </Session>
    </Sessions>
```

```
  <Teams>
    <Team id="HOME Team"/>
    <Team id="AWAY Team"/>
  </Teams>
  <Players>
      <Player id="1" teamId="HOME Team">
          <ShirtNumber>1</ShirtNumber>
      </Player>
      <Player id="2" teamId="HOME Team">
          <ShirtNumber>2</ShirtNumber>
      </Player>
      ...
  </Players>
  <Devices>
    <Device id="dev1">
      <Name>BallJames Ball Tracking</Name>
      <Sensors>
        <Sensor id="position">
          <Name>Position</Name>
          <Channels>
            <Channel id="x">
              <Name>X Position</Name>
              <Unit>m</Unit>
            </Channel>
            <Channel id="y">
              <Name>Y Position</Name>
              <Unit>m</Unit>
            </Channel>
            <Channel id="z">
              <Name>Z Position</Name>
              <Unit>m</Unit>
            </Channel>
            <Channel id="v">
              <Name>Velocity</Name>
              <Unit>m/s</Unit>
            </Channel>
          </Channels>
        </Sensor>
      </Sensors>
    </Device>
  </Devices>
  <PlayerChannels>
    <PlayerChannel id="1_x" channelId="x" playerId="1"/>
    <PlayerChannel id="1_y" channelId="y" playerId="1"/>
    <PlayerChannel id="1_z" channelId="z" playerId="1"/>
    <PlayerChannel id="1_v" channelId="v" playerId="1"/>
    <PlayerChannel id="2_x" channelId="x" playerId="2"/>
    <PlayerChannel id="2_y" channelId="y" playerId="2"/>
    <PlayerChannel id="2_z" channelId="z" playerId="2"/>
```

```
      <PlayerChannel id="2_v" channelId="v" playerId="2"/>
      (...)
    </PlayerChannels>
</Metadata>
<DataFormatSpecifications>
    <DataFormatSpecification separator=":" startFrame="8448" endFrame="165425">
        <StringRegister name="frame"/>
            <SplitRegister separator=";">
            <SplitRegister separator=",">
                <PlayerChannelRef playerChannelId="1_x"/>
                <PlayerChannelRef playerChannelId="1_y"/>
                <PlayerChannelRef playerChannelId="1_z"/>
                <PlayerChannelRef playerChannelId="1_v"/>
            </SplitRegister>
            <SplitRegister separator=",">
                <PlayerChannelRef playerChannelId="2_x"/>
                <PlayerChannelRef playerChannelId="2_y"/>
                <PlayerChannelRef playerChannelId="2_z"/>
                <PlayerChannelRef playerChannelId="2_v"/>
            </SplitRegister>
            (...)
        </SplitRegister>
        <SplitRegister separator=",">
            <BallChannelRef channelId="x"/>
            <BallChannelRef channelId="y"/>
            <BallChannelRef channelId="z"/>
            <BallChannelRef channelId="v"/>
        </SplitRegister>
    </DataFormatSpecification>
</DataFormatSpecifications>
```

Listing B.2: Example of the Source B XML configuration

Table B.1: Example of the TXT file from Source B with the position measurements

```
8448:.000,.000,.000,.000;17.011,57.107,.000,.000;.000,.000,.000,.000; (...)
    :35.776,29.522,.222,.137
8449:.000,.000,.000,.000;17.076,57.092,.000,1.683;.000,.000,.000,.000; (...)
:36.180,29.281,.237,11.773
8450:.000,.000,.000,.000;17.151,57.106,.000,1.894;.000,.000,.000,.000; (...)
:36.524,29.072,.240,10.064
(...)
```